SUSE 10.1 Полная версия

на DVD

Сторона 1
SUSE 10.1
Сторона 2
OpenSource Format
БИТРИКС 5.0
Kororaa XGL.0.2

LINUX FORMAT

ЧИТАЮТ ВСЕ ПОЛЬЗОВАТЕЛИ LINUX

СОЗДАЙ 3D-ИГРУ

Напиши стрелялку на C++ – начни с этого номера! с. 84

Чем Linux круче Vista

3D-интерфейс | Мгновенный поиск | Железная безопасность

Получите все заявленные функции Vista сегодня – в Linux! с. 42

10 Тайных кладов

Горячая десятка приложений, о существовании и необходимости которых вы даже не догадывались! с. 56

№8 (82)
август 2006

Стюарт Коэн

Шеф Линуса Торвальдса защищает право Microsoft присоединиться к OSDL с. 38



ПРИГЛАСИТЕЛЬНЫЕ БИЛЕТЫ
для читателей нашего журнала:
- на SofTool/Linuxland – с. 69
- на LinuxWorld – с. 117

LINUX FORMAT

К вашим услугам...

В этом месяце мы задали Команде LXF следующий вопрос:
Каково, по-вашему, секретное оружие Linux в битве с Vista?



Пол Хадсон
Ну, это просто. Крылья Тукса защищают не хуже стали



Грэм Моррисон
А зачем нам секретное оружие? Vista – по определению, что-то далекое, что отодвигается от нас по мере нашего приближения к нему.



Эфрейн Эрнандес-Мендоса
Любовь... И гранатомет, нацеленный прямо в гордость Билла Гейтса



Майк Сондерс
Я думаю, Linux упрощает основные парадигмы электронной коммерции и обладает динамическим TCO



Ребекка Смолли
Бизнесмены, готовые превратно истолковать MS: все эти фирмы, консультанты и прочие, пытающиеся убедить предприятия попробовать FOSS



Эндрью Грегори
Да кому это надо? Ух, вращающийся куб! Ах, дрожащие окна! Блеск!



Джоно Бэкон
Прозрачность и тот факт, что у нас нет секретов. Если бы они были, я бы ни за что не сказал вам, что по выходным Пол превращается в Паулу. Ой!



Марк Бейн
Секретное оружие? Одно слово – Microsoft. Просто откиньтесь на спинку кресла и ждите еще одной волны переходов на Linux



Нейл Ботвик
Vista – это узкий взгляд на проблему, так что Linux автоматически превращается в решение для тех, кто не носит наглазники. Не говоря уж о том, что для Vista нужна Visa.



Энди Ченнел
Доступность, функциональность, цена, безопасность, приложения, сообщество, LXF, толстый пингвин, гибкость – я могу говорить долго.



Microsoft, айда с нами!

Мы нечасто упоминаем Microsoft на страницах нашего журнала, и не потому, что находимся с ней в жесткой конфронтации – просто формат Linux Format и активное сообщество читателей не дают отвлекаться от основной темы. Однако, в текущем выпуске мы решили нарушить эту традицию. И тому имеются веские причины.

Во-первых, не за горами выход Windows Vista. Сложившаяся обстановка напоминает середину 90-х: в техническом плане Linux уже тогда был способен с конкурировать с однозадачным MS-DOS и Windows 3.1, однако, отсутствие поддержки со стороны крупных компаний вывело на первые позиции совсем другую систему. Сегодня мы имеем ту же ситуацию – разве что за Linux теперь стоят серьезные игроки. В том, что Linux в техническом смысле способен дать фору Vista, сомневаться не приходится, а если вас все же гложет неуверенность – прочитайте спецрепортаж этого номера. Подозреваю, что завидев его самый закоренелый хакер из застенков Microsoft может от огорчения съесть свою накладную бороду.

Кстати, о застенках. Об интерьере зданий Microsoft ходят самые разные слухи. Поговаривают, что в комнатах для отдыха сотрудников там развешаны мишени дартс с портретом Линуса Торвальдса, а под потолком висят камеры пингвинодетекторов. Со свойственной нам прямотой заявляем – либо эти слухи не соответствуют действительности, либо система туксобезопасности Microsoft до сих пор работает под управлением Windows 95. Так или иначе, наш штатный литературный редактор Елена Толстякова, гостившая в Редмонде по личным делам, не только без проблем пронесла в здание Microsoft этот номер Linux Format, но даже отредактировала там большую часть материалов. Да, вы не ослышались – этот номер журнала был подготовлен к печати прямо под носом у Билла Гейтса! И если это как-то отразилось на качестве материалов, то только в лучшую сторону.

Наконец, мы протянули руку помощи нашим товарищам, работающим в глубоком тылу врага. Не имея возможности показать свое истинное лицо, они разрушают проприетарную систему изнутри, подсовывая закоренелым пользователям Windows открытые программы. Специально для них мы разместили на второй стороне DVD сборник открытого ПО для Windows – Open Source Format.

Таким образом, открыв августовский номер LXF, вы приобрели в свой арсенал мощное оружие. Используйте его по своему рассмотрению и... пусть расцветают все цветы.



С уважением,
ВАЛЕНТИН СИНИЦЫН
главный редактор LinuxFormat в России

КАК С НАМИ СВЯЗАТЬСЯ

Письма для публикации:
letters@linuxformat.ru
Подписка и предыдущие номера:
subscribe@linuxformat.ru
Техническая поддержка:
answers@linuxformat.ru
Проблемы с дисками:
disks@linuxformat.ru
Общие вопросы: **info@linuxformat.ru**
Website: **www.linuxformat.ru**

Адрес редакции: Россия,
Санкт-Петербург, ул. Гончарная, 23, офис 54
Телефон редакции: (812) 717-00-37

Дополнительная информация на стр.134

МИССИЯ ЖУРНАЛА

- Пропаганда свободного ПО в России
- Продвижение решений с открытым кодом в бизнес-сообществе
- Поддержка российского Open Source сообщества
- Организация трибуны для разработчиков свободного ПО
- Обратная связь между разработчиками и потребителями ПО

Corporate Patron Free Software Foundation

Содержание

Добро пожаловать в *LinuxFormat* – ваш гид в мире Linux!

LINUX FORMAT

LXF8(82) АВГУСТ 2006





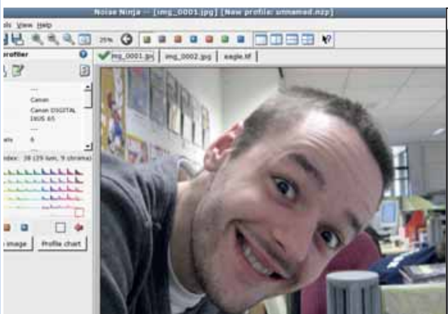
МЫСЛИ ВСЛУХ

«SymphonyOS, несомненно, самая оригинальная из ныне существующих настольных ОС.»

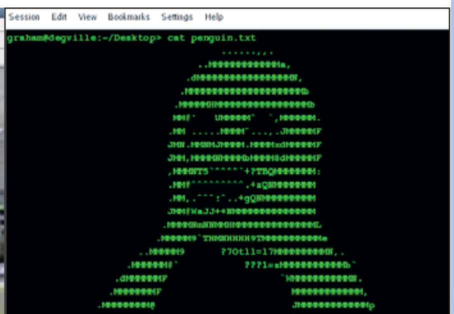


«Оно не хотел говорить об этом. К счастью, никто не может устоять перед LXF!»





10 Фото атакует



16 Лучший терминал

На диске

Целый DVD интересных программ 128



» DVD!

SUSE 10.1 Полная версия дистрибутива – установите ее на домашний компьютер, рабочую станцию или сервер

УЧЕБНИКИ PHP 200 страниц руководств, советов и прочего из LXF30-82.

AMAROK 1.4 Управляйте своей аудиотекой вместе с новым музыкальным плеером для KDE

SYLLABLE 0.6.1 Уникальная открытая настольная ОС, быстрая и простая в использовании – попробуйте!

MANIADRIVE Достойный продолжатель традиций Stunt Racer. Попахивает жженой резиной!





Перед использованием диска ознакомьтесь с инструкцией и вырежьте обложку своего DVD на стр. 131–132



Главные новости

• Qt 4.2 TP 1 • NTFS на чтение и запись • Курсы Linux в двух столицах • Microsoft OpenDocument Format

Qt 4.2 Technology Preview 1

Интеграция, интеграция и еще раз интеграция

Норвежская компания Trolltech (www.trolltech.com) объявила о выпуске *Qt 4.2 Technology Preview* (TP) 1 – первой тестовой версии нового поколения популярной библиотеки для программирования GUI. По сравнению со своими предшественницами, *Qt 4.2* содержит значительное количество улучшений, среди которых хочется особо отметить работу в области интеграции.

Как известно, одной из первоочередных задач *Qt* является корректная эмуляция внешнего вида и поведения приложений, принятого на той или иной платформе. Если в отношении Windows или Mac OS X вопросов не возникает, то в Linux дела обстоит несколько иначе. В этой системе понятие «родное окружение» включает в себя как построенную на *Qt* среду KDE, так и использующий *Gtk+* рабочий стол Gnome. Естественно, приложения *Gtk+* и *Qt* выглядят и ведут себя несколько по-разному, однако, выход финальной *Qt 4.2* может в значительной степени нивелировать это отличие.

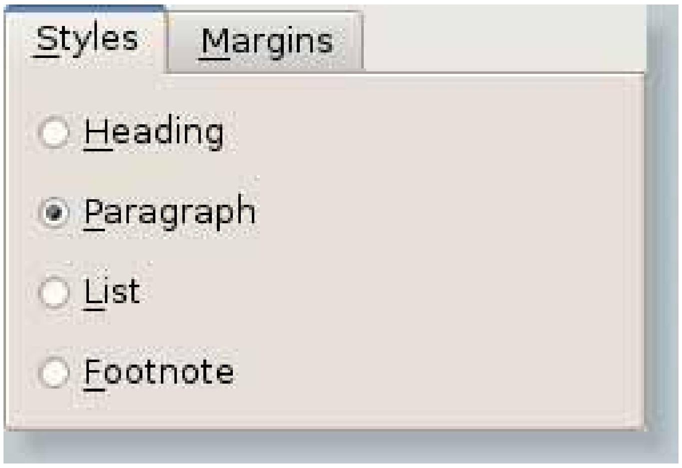
Начнем с того, что бросается в глаза – внешнего вида. Одним из возможных подходов здесь является использование универсальных визуальных тем вроде *QtCurve*, однако, *Qt 4.2* предлагает альтернативный вариант – встроенную тему Cleanlooks, эмулирующую де-факто стандартный внешний вид современных приложений Gnome. Добавьте к этому родной порядок следования кнопок в диалогах Gnome (благодаря соответствующему расширению класса QDialogButtonBox) и вы едва ли сможете отличить *Gtk+* от *Qt* «на глаз».

Впрочем, интеграция между двумя ведущими инструментариями имеет куда более глубокие корни. Теперь приложения Gnome допущены в «святая святых» – цикл сообщений *Qt*. Интеграция с циклом сообщений Glib, реализованная в Qt 4.2, позволит например, использовать в приложениях Qt «чужие» подключаемые модули, и наоборот – использовать подключаемые модули Qt в «чужих» приложениях.

В то время, как рабочая группа проекта Portland только разрабатывает универсальный механизм открытия web-страницы, запуска почтового клиента и так далее, *Qt 4.2* уже предлагает готовый класс *QDesktopServices*, предоставляющий кросс-платформенный способ для показа различных ресурсов, идентифицируемых по URL.

Впрочем, межпроцессное взаимодействие никоим образом не ограничивается запуском браузера с последующим перенаправлением пользователя по указанному адресу. В среде Linux все большую силу набирает D-BUS (http://www.freedesktop.org/wiki/Software/dbus) – средство для обмена сообщениями, используемое HAL, NetworkManager и другими настольными технологиями. Приложения *Qt* могли использовать D-BUS еще в третьей версии библиотеки (правда, тогда речь шла только об интеграции с циклом сообщений *Qt* – интерфейс D-BUS был выдержан в стиле C и не соответствовал «Qt way»), однако, «привязки» D-BUS стали неотъемлемой частью *Qt* только в версии 4.2. На сей раз в распоряжение программиста предоставлен «правильный» C++ интерфейс в стиле *Qt*. С учетом произошедшего несколько месяцев назад портирования kdelibs с DCOP на D-BUS, можно с уверенностью утверждать, что это нововведение придется весьма кстати.

Ну и наконец скажем пару слов о системном лотке (system tray). Классы для работы с ним существовали в KDE с начала времен, но, по вполне понятным причинам, область их применения была ограничена рамками X. *Qt 4.2* предлагает вашему



Ubuntu? Нет, Qt 4.2!

вниманию полностью переносимый класс *QSystemTrayIcon*, который позволяет разместить иконку в трее способом, принятым на целевой платформе.

Пройдет не так много времени и находящаяся сейчас в стадии предварительной версии *Qt 4.2* ляжет в основу графического окружения KDE4. Обозначенные выше функции позволяют надеяться, что как только это случится, разнородный рабочий стол Linux займет свое почетное место в истории этой ОС, уступив место крепко сбитому интегрированному рабочему окружению.

C++ GUI PROGRAMMING WITH QT 4

В середине июля было официально объявлено о выходе в свет книги «C++ GUI Programming with Qt 4», получившей официальное одобрение Trolltech и написанной ее сотрудниками: Жасмин Бланшетт [Jasmin Blanchette] и Марком Соммерфильдом [Mark Summerfield], правда, последний некоторое время назад покинул компанию и основал собственное дело. Книга, по сути, является расширенной версией «C++ GUI Programming with Qt 3», включенной в серию Bruce Perens Open Source Series (phptr.com/perens) и изданной в России издательством «Кудиц-Образ». Текст не претерпел существенных изменений – вы найдете здесь те же главы и примеры, что и в предыдущей версии учебника, адаптированные к Qt4. Существенно новыми являются пара глав, посвященных ключевым нововведениям Qt4: MVC, Arthur, подключаемым модулям. Плюс это или минус, решайте сами. Интерес представляет также схема распространения книги. Она не является частью Bruc Perens Open Source Series, однако, надпись на форзаце гласит, что ее текст распространяется на условиях Open Publication License, v1.0 или выше. Найти электронную версию учебника в официальных источниках пока не удалось, возможно, она будет опубликована после продажи некоторого количества «бумажных» экземпляров.

http://www.phptr.com/bookstore/product.asp?isbn=0131872494&rl=1

NTFS на запись и на чтение

В Linux наконец-то появился полноценный драйвер для файловой системы Windows XP

Похоже, что извечная проблема совместного использования Windows и Linux на одном компьютере – запись на разделы *NTFS*, наконец-то нашла свое решение. Сакашиц Шаболч [Szakacsits Szabolcs] анонсировал выпуск бета-версии драйвера *ntfs-3g*, обеспечивающего устойчивую поддержку *NTFS* в режиме чтения-записи.

Строго говоря, *ntfs-3g* не является драйвером в полном смысле этого слова. Это программа, работающая в пространстве пользователя и предоставляющая свои функции через FUSE. Впрочем, конечному пользователю до этого – какое дело? Ему важно, что теперь нет никакой необходимости содержать FAT32-раздел с исключительной целью обмена данными между Windows и Linux. Кстати, производительность от использования FUSE отнюдь не страдает – согласно проведенным автором тестам, *ntfs-3g* существенно обгоняет как свободный Captive NTFS, так и коммерческий Paragon NTFS и уступает только родным для Linux JFS/ReiserFS (*см. таблицу*). И это, как говорится, не предел – разработчики утверждают, что производительность можно повысить аж в несколько раз. Стабильность работы *ntfs-3g* также на высоте – драйвер был протестирован на более чем 40 снимках реальных NTFS-разделов и ни разу не привел к краху ФС.

Технически, *ntfs-3g* является улучшенной версией программы ntfsmount, разрабатываемой в рамках проекта Linux-NTFS. Он не поддерживает доступ к зашифрованным файлам, запись в сжатые файлы, а также изменение имени владельца и прав доступа. Исходные тексты *ntfs-3g* лицензированы под GPL и, на момент написания этих строк, были доступны по адресу: **http://mlf.linux.rulez.org/mlf/ezaz/ntfs-3g-20070714-BETA.tgz**.

Усредненные данные по скорости создания, удаления и доступа к файлам получены с помощью 'bonnie++ -s0', запускаемой на различных ФС. Число файлов на каталог в каждом случае составляло 16 000

	files	/sec	%CP	/sec	%CP	/sec	%CP	/sec	%CP	/sec	%CP	/sec	%CP
reiserfs	16k	21459	99	+++++	+++	17856	96	20172	98	+++++	+++	16414	96
jfs	16k	7015	13	+++++	+++	5868	10	3068	14	+++++	+++	1075	3
ntfs-3g	16k	3021	99	14291	99	5226	99	3548	99	16149	99	5223	99
xfs	16k	2401	17	+++++	+++	2095	15	2301	20	+++++	+++	347	2
ext3	16k	1862	96	+++++	+++	+++++	+++	1914	96	+++++	+++	9695	98
minix	16k	1450	97	+++++	+++	18148	94	1694	97	+++++	+++	4847	98
fat32	16k	366	97	+++++	+++	1809	97	428	97	+++++	+++	1361	97
paragon ntfs	16k	58	98	1259	99	245	99	55	99	+++++	+++	832	99
captive ntfs													

Microsoft Office становится ближе к ODF

Корпорация Microsoft запустила открытый проект по созданию конвертера Open XML-ODF

Корпорация Microsoft, долгое время находившаяся в оппозиции к *OpenDocument* и продвигающая собственный формат для офисных документов – Open XML, вняла просьбам своих клиентов (особенно усердствовали правительственные учреждения) и запустила проект, целью которого будет создать инструмент для конвертации между Open XML и *OpenDocument*. Продукт будет бесплатным и доступным в качестве дополнения (add-in) для старых версий Microsoft Office, более того, его разработка будет вестись в соответствии с принципами Open Source на сайте **SourceForge.net**.

Конвертер, получивший гордое название Open XML Translator, будет разрабатываться совместно с французской компанией Clever Age, а также Aztecsoft (Индия) и Dialogika (Германия). Прототип приложения для Word 2007 уже опубликован по адресу **http://sourceforge.net/projects/odf-converter** и распространяется по лицензии BSD. Финальная версия будет доступна к концу 2006 года, сообщает Microsoft.

ЧТО БЫЛО

★ Летняя школа Linux в Обнинске
22-24 июля 2006 года, Обнинск

В период с 22 по 24 июля в Обнинске (Калужская область) проходила «Летняя школа Linux», организаторами которой выступили компания ALT Linux и Интернет-Университет Информационных Технологий. В Школе можно было изучить основы Linux и познакомиться методикой его преподавания (на базе курса «Операционная система Linux»). Кроме этого, в рамках Школы прошел семинар, посвященный легальной миграции на Linux.

★ Восьмой ежегодный фестиваль Linux в Калужской области
28-30 июля, р. Протва

Восьмой ежегодный фестиваль Linux в Калужской области проходил в рамках «Недели Линукс На Протве» с 28 по 30 июля 2006 года в Боровском районе Калужской области. Фестиваль, как и его предшественники, стал местом встречи единомышленников «под открытым небом». Команда LXF, занятая подготовкой этого номера к печати, была, к сожалению, вынуждена пропустить это мероприятие, но мы не теряем надежды когд-нибудь выбраться из душного города на берега живописной реки. В палатки.

ЧТО БУДЕТ

★ LinuxLand / SofTool'2006
26-29 сентября 2006 года, Москва

Компании ИТ-Экспо и LinuxCenter.Ru приглашают вас принять участие в выставке информационных технологий SofTool'2006 (26-29 сентября 2006 года), где планируется собрать ведущие российские Linux-компании в одном секторе выставочной площади LinuxLand. LinuxLand будет местом, в котором соберутся поставщики Linux и различных решений для этой ОС: Mandriva, IBM, Novell, R-Style, HP, Oracle, ASPLinux, Linux-Online (разработчик Linux XP), НПО "Сеть" (разработчик MOPSLinux), Bitrix, ПРОМТ, Etersoft и Linuxcenter.ru, журнал Linux Format, образовательный центр Lynx Education Center и другие. Помимо выставочных стендов, на экспозиции традиционно будет расположена демо-зона, где посетители LinuxLand смогут вживую познакомиться с предлагаемыми продуктами. В ходе выставки будут подведены итоги конкурса IBM WebSphere Community Edition Contest 2006.
www.linuxland.ru

Курсы Linux в двух столицах



Теперь пройти обучение по программе Linux Professional Institute стало возможным не только в Санкт-Петербурге, но и в Москве

География чтения авторизованных курсов Linux Professional Institute (LPI) расширилась с получением учебным центром R-Style (Москва) нового звания «Авторизованный УЦ Mandriva Linux». Это позволит R-Style проводить обучение по программам, разработанным специалистами Mandriva и сертифицированным независимой организацией – Linux Professional Institute. Ранее подобные курсы читались только на базе LYNX Education Center в городе Санкт-Петербурге.

Особенность курсов LPI состоит в их «нейтральности», то есть в отсутствии ориентации на конкретный дистрибутив. Например, тот факт, что курсы, читаемые в УЦ R-Style, разработаны компанией Mandriva, не означает, что их содержание будет сведено к фирменным инструментам, предлагаемым именно этой организацией. Напротив, слушатели получат информацию об общем устройстве Linux, не привязываясь к специфике конкретного поставщика.

Программа обучения LPI построена таким образом, чтобы охватить все уровни начальной подготовки слушателей. Начинающий сможет поближе познакомиться с новой ОС, а опытный пользователь – повысить свой уровень. Предлагаемые курсы включают в себя:

- **LNX70** – базовые концепции и методы использования основных средств Linux;
- **LPI101** – базовый курс по системному администрированию, установке и управлению отдельной Linux-системой;
- **LPI201** – дополнительные вопросы системного администрирования, установки и управления отдельной Linux-системой;
- **LPI102** – расширенный курс по системному администрированию, установке и управлению отдельной Linux-системой;
- **LPI202** – дополнительные вопросы сетевого администрирования Linux-системы.

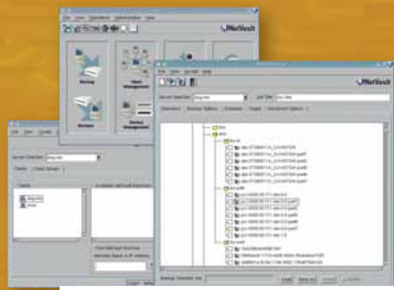
Со временем, авторизованные учебные центра Mandriva/LPI будут открыты и в других крупных городах России. LXF

КОРОТКОЙ СТРОКОЙ

- Вышла в свет финальная версия SUSE LINUX Enterprise Server/Desktop 10.
- Проект Freespire представил на суд общественности первую бета-версию community-редакции Linspire на две недели раньше намеченного срока
- После продолжительного затишья и смены лидера проект SIM-IM выпустил новую версию популярного IM-клиента: 0.9.4
- Корпорация IBM представила Linux-версию пакета Lotus Notes, базирующуюся на Eclipse.
- В Алматы (Казахстан) открылся собственный Open Source центр. Проект нацелен на снижение уровня использования нелицензионного программного обеспечения и популяризацию открытого ПО
- Проект KDE и компания Apple начали работу по синхронизации WebKit и Konqueror, а компания Frologic выпустила бесплатную версию тестера Squish для KDE-приложений.

Обзоры >>

Новинки программного и аппаратного обеспечения в описании наших экспертов



12 Резервное копирование и даром, и задорого

СЕМЬ ШАГОВ LINUX-ДИСТРИБУЦИИ



Алексей Федорчук Имеет собственное мнение насчет финансирования OSS.

Первые дистрибутивы Linux, возникли из стремления избавиться от лишней ОС – ведь, чтобы сварить суп из курицы (Linux), следовало как минимум иметь кошку (MINIX). И, например, спасение пользователей от ада пакетных зависимостей оставалось делом рук самих пользователей – благо таковыми в то время были почти исключительно разработчики Linux'а же.

Потом началась эпоха промышленного применения этой ОС – для начала в качестве сетевых узлов разного рода. И к пользователям-разработчикам присоединились пользователи-администраторы, которые не имели времени на ручное разруливание зависимостей – и для них были придуманы первые дистрибутивы с контролем оных (*Debian, Red Hat*).

В 1998 году впервые заговорили о продвижении Linux на пользовательские компьютеры. Итогом их стало появление *Mandrake* – первого по-настоящему «юзерофильного» дистрибутива.

Однако скоро пользователи осознали, что на своих рабочих столах они являются также и администраторами, что вызвало волну популярности дистрибутивов *Source Based* – и пальму первенства пользовательских симпатий завоевал *Gentoo*. Каковой тоже не стал панацеей от всех бед – потребовались системы, совмещающие возможность полной пересборки с быстротой развертывания и простотой обновления – квинтэссенцией этого направления стал *Archlinux*.

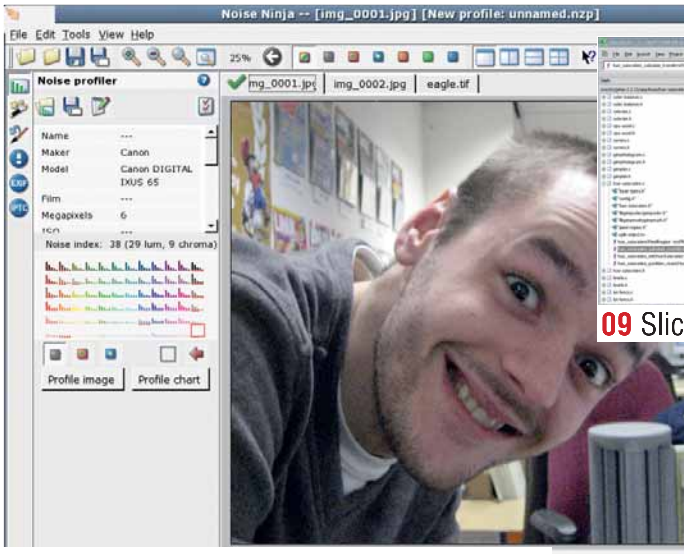
Дальнейшее стремление к упрощению жизни вернуло интерес к *Debian* и его многочисленным клонам, среди которых бесспорно первенствовал Ubuntu.

И, наконец, нынче намечается обращение к истокам Linux-дистрибуции – в лице современных производных *Slackware*, в первых рядах которых выступает *ZenWalk* – максимально компактный, но легко наращиваемый.

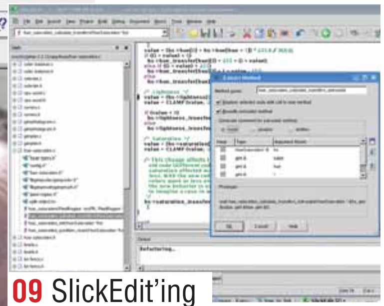
Каким будет следующий виток пользовательских предпочтений?

alv@posix.ru

ЧТО НОВОГО?



10 Легким движением руки...



09 SlickEdit'ing

09 SLICKEDIT 11

Мы всегда с недоверием относились к текстовым редакторам для программистов, особенно с номерами версий больше 10. Стоит ли он своих денег?

10 NOISE NINJA 2.1

Ненавидите шум, но любите ниндзя? Этот прекрасный инструмент для фотографов теперь имеет порт и для Linux!

11 FREEBSD 6.1

Вы, возможно, читали введение во FreeBSD, которое Майк опубликовал в LXF76. Он снова с вами и готов рассмотреть последнюю версию этой ОС!

12 AMANDA 2.5 VS NETVAULT 7.4

Бесплатное резервирование против платного – и неожиданный результат. Смотрите сами!

НАШ ВЕРДИКТ: ПОЯСНЕНИЕ

Все продукты оцениваются по 10-бальной шкале. 4 обычных параметра оценки: возможности, производительность, простота использования и соотношение «цена/качество», но для свободного ПО последний параметр может быть заменен на оценку документации. Независимо от набора категорий, мы всегда вычисляем общий рейтинг, подводящий итог нашим высказываниям.



Продукты, выделяющиеся из основной массы, получают престижную награду *Linux Format Top Stuff Award*. Выбираются самые-самые – и только лишь высокая оценка здесь еще ни о чем не говорит.

Для тестирования серверов используется наша собственная разработка, LFXBench 2004, состоящая из 4 основных подсистем: Multi CPU, Single CPU, RAM и Hard Disk. Оценки усредняются и дают общий рейтинг.

1 очко означает, что тестовый компьютер равен нашему эталону: Pentium 4 1,8ГГц, 512 Мб оперативной памяти и жесткий диск IDE. 2 очка означает, что он вдвое быстрее.

Все тесты выполняются под управлением Red Hat Enterprise Linux 3 AS на соответствующей платформе (x86, AMD64, Itanium). Код компилируется при помощи GCC, если не указано иное.

РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ

MULTI CPU	6.07
SINGLE CPU	3
RAM	2.17
HARD DISK	0.46

ОБЩИЙ РЕЙТИНГ 2.93

ТЕКСТОВЫЙ РЕДАКТОР

SlickEdit 11

Ник Вейч размышляет, стоит ли платить за право кодировать.

САМОЕ ГЛАВНОЕ

Очень гибкий редактор для программистов с поддержкой нескольких языков программирования. Аналоги: *Kate*, *Emacs*, а также среды разработки типа *Eclipse*.

- **РАЗРАБОТЧИК:** SlickEdit
- **САЙТ:** www.slickedit.com
- **ЦЕНА:** $284 для одного пользователя, $139 за обновление

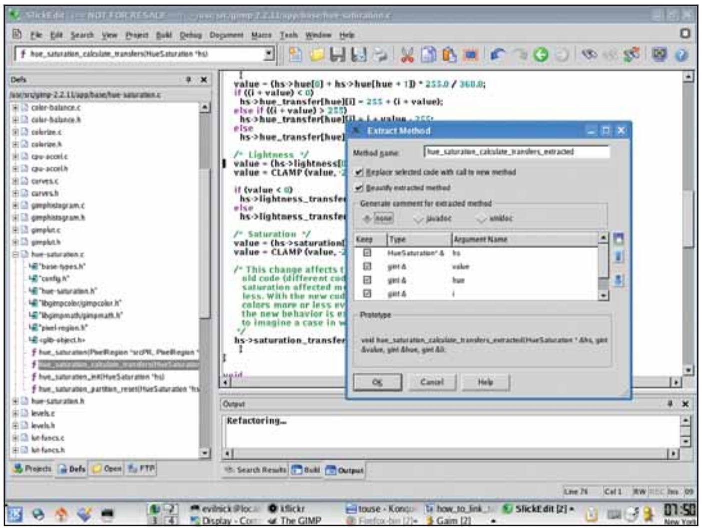
Программисты в Linux избалованы выбором. *KDevelop*, *Anjuta*, *Eclipse*... множество сред разработки для написания, компиляции и отладки вашего кода, и в основе каждой – текстовый редактор. Так зачем мучиться с обычным текстовым редактором, создавая новейший алгоритмический шедевр, если можно воспользоваться специально созданным инструментом?

Редакторов для программистов на свете хватает: взять хотя бы *Emacs*, *Kate*, или *JEdit* – да мало ли добротных бесплатных инструментов! Поэтому платная программа должна обладать прямо-таки супер-качеством, чтобы ее вообще заметили. К счастью для разработчиков, *SlickEdit* имеет солидную репутацию хорошо оснащенной программы, и настоящая версия добилась еще больших успехов.

Со времени нашего последнего обращения к *SlickEdit* включены некоторые новые функции и, что не менее важно, усовершенствованы старые. Наиболее заметные из них – поддержка PHP 5, функция автогенерации для файлов Javadoc и XMLDoc, улучшенные инструменты регулярных выражений, поиска и замены, автозавершения кода и закладок. Подарок для Linux-версии – дополнительный пакет поддержки шрифтов *Xft* через *fontconfig*.

Одно из замечательных нововведений – новый, быстрый режим рефакторинга кода (Quick Mode). Ранее эта система была очень устойчива и надежна, но уж больно медленна – даже для обычной смены имени переменной или подобных простых вещей. Новый режим обеспечивает замену имен с помощью тэг-файла для поиска всех вхождений выделенного текста – версия 11 это заметно ускорила, но требует известной осторожности в сложных ситуациях, например, при потенциальном конфликте перегруженных операторов. Среди функций быстрого рефакторинга – Method Extract, подсветка участка кода для автоматического выделения его как функции. Дополнения PHP 5 тоже пришлись ко двору, получить бы только правильную подсветку новых вызовов, например, *mysqli_connect()*.

Контроль версий *SlickEdit* поддерживает и *Subversion*, и CVS, и кучу других систем. По правде говоря, это просто ссылки на подготовленную командную строку, но гибкость *SlickEdit* такова, что можно создавать и свои собственные, назначением любых команд таким действиям, как Check In и History. Эта гибкость характерна для всей программы. В десятках диалоговых окон настройки можно менять сочетания клавиш (которые вам, вероятно, захочется отредактировать, хотя по умолчанию *SlickEdit* предлагает эмуляцию стилей *Vi*, *Emcs* и др.), цвета кода, подсветку синтаксиса, расширения файлов, отступы, шрифты и т.д.



Рефакторинг простых операторов стал быстрее – но будьте с ним поосторожнее!

Для программистов

Конечно, одна из причин выбора специализированного редактора вроде этого – богатство инструментария. Подсветка синтаксиса – такая же необходимость для сегодняшних профессиональных программистов, как автозавершение кода (которое всегда было проблемой для разработчиков). В этом отношении *SlickEdit* хорош, но при наборе участков кода постоянно возникает легкая (и досадная) задержка перед появлением окна автозавершения. Функция, однако, действует безупречно, а «задумчивости» можно поубавить, ограничив завершение только конкретными видами кода.

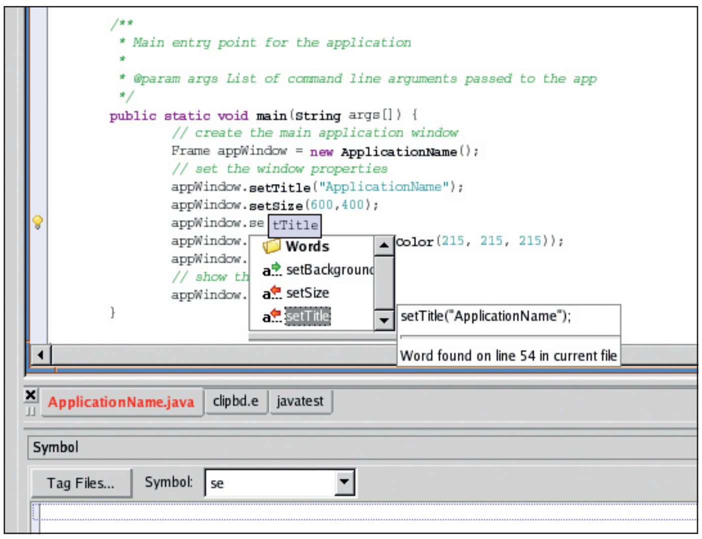
Есть и другие помощники. Если честно, то встроенный калькулятор, хоть и прекрасно интегрированный, малость неуклюж. А вот инструмент работы с регулярными выражениями превосходен – помимо помощи в построении выражений, он поддерживает тестирование правильности их работы.



ГРЭМ СЧИТАЕТ...

«Мне нравится интерфейс пользователя, хоть он и съедает многовато циклов процессора. Однако всем этим функциям далеко до дерева откатов Vim».

Единственный, но жирный минус – внешний вид программы. Понятно, что вкусы программистов различаются, но интерфейс в стиле *Motif* с плоховатыми элементами управления и неудобным основным окном изрядно затрудняет работу с несколькими файлами. Кое в чем *SlickEdit* отстает даже от более специализированных сред разработки для Java. Для обычного корпоративного программиста *SlickEdit* неплох, но учитывая его цену, вы, вероятно, предпочтете программировать в *Kate* или *Emacs* и выполнять рефакторинг вручную. LXF



Автозавершение кода «тормозит», но действует безотказно.

НЕ ЗАБУДЬТЕ О МОДУЛЕ

Если вы Java-программист, то, вероятно, уже пользуетесь превосходной средой разработки Eclipse. Но SlickEdit (фирма) разработала версию своей программы в виде модуля для Eclipse, так что вы можете получить функциональность *SlickEdit*, интегрированную в вашу среду разработки, за более низкую цену ($199) по сравнению с самостоятельной программой.

ВЕРДИКТ LINUX FORMAT

ФУНКЦИОНАЛЬНОСТЬ	9/10
ПРОИЗВОДИТЕЛЬНОСТЬ	9/10
ПРОСТОТА ИСПОЛЬЗОВАНИЯ	7/10
ОПРАВДАННОСТЬ ЦЕНЫ	6/10

Немного неказист, но хорошо оснащен для всех видов работ программиста.

РЕЙТИНГ 7/10

ИНСТРУМЕНТ ФОТОГРАФА

Noise Ninja 2.1

Процесс профессиональной обработки фотографий стал чуточку ближе к **Полу Хадсону**.

САМОЕ ГЛАВНОЕ

Удаляет помехи с цифровых фотографий. Попытайте удачи также в *Gimp*.

- **РАЗРАБОТЧИК:** PictureCode
- **САЙТ:** www.picturecode.com
- **ЦЕНА:** От $34,95

Фотография – не только искусство, но и умение найти баланс технических параметров. Например, для повышения глубины резкости необходимо выставить малую диафрагму и длительную выдержку – но увеличение выдержки обычно приводит к размыванию изображения. Выдержку можно уменьшить вдвое с каждым очередным номиналом ISO-рейтинга (100, 200, 400 и т.д). Однако увеличение рейтинга неминуемо увеличивает электрический шум – разноцветные зерна и пятнышки, производимые фотоэлементом камеры. Изображение, прекрасно выглядящее на ISO 100, на ISO 3200 покроется радужными разводами, сведя на нет все труды по тщательной подборке баланса.

На помощь приходит *Noise Ninja*: смело повышайте ISO-рейтинг, помехи на ваших фотографиях ликвидируются цифровыми методами.

Noise Ninja имеет опыт работы на Mac OS X и Windows, но первый порт для Linux появился лишь в этой версии. Технология отработана: можно избавляться от помех и накладывать сглаживающие маски легким движением руки, а программа сама исправит ваши фотографии.

На Mac OS X и Windows есть выбор: можно приобрести программу как самостоятельное приложение, так и в виде модуля к *Adobe Photoshop*. А так как *Photoshop* для Linux не существует в принципе, то варианты отсутствуют, вы можете получить только самостоятельный продукт. Требовать модуля для *Gimp* преждевременно – вероятно, это дело будущего.

Не имея функциональности модуля, вы ограничены в выборе типов файлов. Например, в *Photoshop* для OS X можно загрузить и обработать фотографию в *Noise Ninja* в формате RAW, а на Linux доступны лишь TIF и JPEG. Имеется поддержка 16-битных TIF (если вы приобрели версию Pro за $70), но из *Gimp* они не экспортируются, и большинство пользователей будет довольствоваться 8-битным цветом. Зато добрые старые JPEG *Noise Ninja* чистит отлично.

> «МОЖНО УБРАТЬ ПОМЕХИ И НАЛОЖИТЬ МАСКУ ОДНИМ ДВИЖЕНИЕМ.»

Базовые функции удаления помех хороши для новичков, но в арсенале *Noise Ninja* есть и более тонкие инструменты. Мы добились прекрасных результатов следующим способом: до отказа выдвинули ползунки Smoothness и Strength (Гладкость и Интенсивность) и получили нереалистично гладкое изображение. Затем постепенно снижали уровни, пока зерна не появились снова.



Простой интерфейс наводит на мысль, что работать в *Noise Ninja* проще простого, но не ждите, что она совершит чудо со сложным изображением!

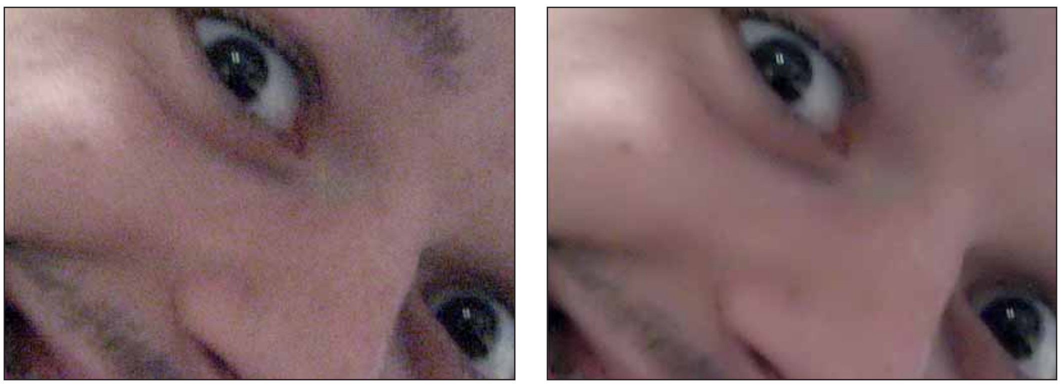
Ползунки и кнопки

Можно просматривать отдельные каналы и определять, где сосредоточены основные помехи – особенно часто этим грешат каналы синего цвета. Если какие-то пятна вам захочется оставить, можно «нарисовать» первоначальное изображение поверх отфильтрованной версии с помощью инструмента Noise Brush (Шумовая Кисть). Он подойдет для восстановления мелких деталей, которые *Noise Ninja* заутюжил чересчур агрессивно.

А вот пакетный процесс – гениальная вещь, это автоматическая фоновая обработка целых директорий с фотографиями. Поскольку удаление помех и фильтры настраиваются раздельно, резкость можно увеличить, скажем, в *Gimp*, а *Noise Ninja* поручить индивидуальную обработку помех для каждой фотографии. На практике это означает, что вы можете одновременно чистить фотографии ISO 100 и ISO 3200: в процессе работы *Noise Ninja* подстроится автоматически. Владельцев мощных компьютеров ждет приятная новость: *Noise Ninja* полностью поддерживает многопроцессорную и многоядерную архитектуры; есть возможность размещения директории с файлами на отдельном разделе для повышения производительности.

Чудо одним щелчком

Каждый фотограф сталкивался с проблемой помех, и *Noise Ninja* – именно то, что вам необходимо для решения этой проблемы (пробная версия всегда наготове на сайте PictureCode). Нам она уже помогла спасти сотни фотографий, и, надеемся, в будущем спасет еще больше – она на это способна.



До удаления помех (слева)**, зерна отчетливо видны; после удаления** (справа) **фотография заметно очистилась.**

ВЕРДИКТ LINUX FORMAT

Простая в изучении, изощренная в работе – без такой вещи не обойтись любому серьезному фотографу.

РЕЙТИНГ 9/10

ОТКРЫТАЯ ОПЕРАЦИОННАЯ СИСТЕМА

FreeBSD 6.1

Команда FreeBSD надеется вернуть своей ОС знаменитую стабильность. **Майк Сондерс** знакомится с новейшей версией.

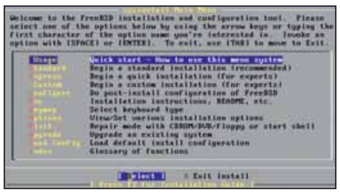
САМОЕ ГЛАВНОЕ

Открытая операционная система из семьи Berkeley Standard Distribution. Аналоги – NetBSD, OpenBSD и, конечно же, Linux!

- **РАЗРАБОТЧИК:** The FreeBSD Foundation
- **САЙТ:** www.freebsd.org
- **ЦЕНА:** Бесплатно по лицензии GPL

Пользователи FreeBSD – публика консервативная. Они не любят скоропалительных, непродуманных добавок в исходные тексты каждой новой версии, как и вставок непроверенного кода в основное дерево, заставляющих пользователя тратить время на отлов ошибок.

Поэтому выпуск версии 5.0 в 2003 г., когда многие разработчики и пользователи сочли систему нестабильной, нанес репутации FreeBSD ощутимый удар. Выпуск 5.0 затянулся на годы, так что, казалось, версии 5.x мы не увидим никогда. Команда FreeBSD выпихнула новую версию, намереваясь доработать и стабилизировать будущие 5.x, но цели не достигла – пресловутая 5.0 наложила отпечаток на все 5.x, и большинство FreeBSD-пользователей остались на 4.x.



Sysinstall **на вид несколько староват, зато инсталляция проходит со свистом.**

Вычищенную FreeBSD 6.0 приняли доброжелательно, но многие продолжали ждать следующей точечной версии, надеясь, что уж в ней-то FreeBSD точно «устаканится». Поэтому 6.1 – эволюционная версия: в ней нет ни крупных перемен, ни новых функций, только подчистка и доводка. Написаны новые стресс-тесты для проверки стабильности штатной файловой системы FreeBSD, UFS, имеющей функции журналирования в лице SoftUpdates. Поддержка Bluetooth была усовершенствована автоматической настройкой многих устройств, включение поддержки сетевых карт Broadcom NetXtreme II, AMD Am7900 LANCE и Am79C9x расширило набор сетевых драйверов. Текущая версия FreeBSD работает на платформах x86 PC, AMD64, Itanium и SPARC и занимает два диска – первый несет собственно систему, а на втором разместились настольные и серверные пакеты.

Хотите приятных известий? Наши тесты показали полный возврат 6.1 к уровню стабильности 4.x. Тяжелую нагрузку система несет уверенно, оставаясь отзывчивой к командам даже в периоды высокой активности процессора и жесткого диска.

Одно из интересных для конечных пользователей новшеств – это Portsnap, позволяющий скачивать сжатые снимки дерева портов (система построения ПО FreeBSD, вдохновившая *Portage* под *Gentoo*). Это помогает держаться в курсе последних событий – ценное дополнение для любителей новейших версий программ. Пользователи FreeBSD обновляют свои системы заплатками на исходный код из CVS; это медленнее бинарных обновлений Debian, зато дает полный контроль над процессом компиляции.



Рабочие столы Linux и FreeBSD почти неотличимы – у них одинаковы основная среда и большинство популярных приложений.

Рабочий стол FreeBSD 6.1 включает *X.org 6.9.0*, *Gnome 2.12.3* и *KDE 3.5.1*, с серверной стороны – *MySQL 5.0.18*, *Apache 2.2.0*, *Sendmail 8.13.6* и *Bind 9.3.2*.

Итоги

Sysinstall, текстовый инсталлятор, не изменился по сравнению с 6.0. Конечно, с виду он не так изобретателен, как, например, Anaconda или Yast, зато здорово ускоряет инсталляцию (меньше 10 минут для базовой системы вместе с X на нашей 2 ГГц тестовой машине). Время загрузки тоже на шаг впереди большинства Linux-дистрибутивов – 40 секунд до экрана входа в систему против 73 для SUSE 10.1.

Пользуетесь ли вы серверным или настольным вариантом – FreeBSD почти неотличима от Linux. В коллекции Ports – свыше 14 000 программ (сравните с 15 500 для Debian), среди которых найдется почти любое приложение, название которого вы только можете вспомнить. Раскладка файловой системы FreeBSD почти такая же, как и в Linux, за исключением нескольких микроразличий – например, все основные загрузочные параметры собраны в **/etc/rc.conf**, а не разбросаны в разных местах, как в Linux. Если вы попробуете установить FreeBSD с нуля, имея опыт работы в Linux, то вряд ли наткнетесь на какие-либо сюрпризы, но в случае затруднений всегда можно обратиться к онлайн-учебнику FreeBSD Handbook – блестяще написанному централизованному ресурсу для всей FreeBSD, до мельчайших ее деталей. [Похвала ничуть не преувеличена, к тому же учебник доступен и в виде архива для автономного чтения, – прим. пер.] Эта версия будет поддерживаться как минимум 12 месяцев.

Если вы счастливый обладатель 6.0, то рвануть на 6.1 особой нужды нет. Изменения минимальны – как раз в стиле пользователей FreeBSD. Если вы никогда еще не пробовали FreeBSD, сейчас самое время оценить ее непревзойденную стабильность, дружелюбный дизайн и первоклассную документацию. А благодаря режиму совместимости с Linux большинство бинарных дистрибутивов Linux-программ прекрасно уживаются с FreeBSD, и вам не придется жертвовать своей коллекцией приложений. **LXF**

ГДЕ РАБОТАЕТ FREEBSD?

Linux, конечно, любимец компьютерной прессы, но FreeBSD достойно несет трудовую вахту на многих ответственных постах. Например, Yahoo использует ее на большинстве своих серверов, Hotmail тоже была привержена FreeBSD до вхождения в Microsoft. Именно на серверной территории FreeBSD нашла себя: администраторы ценят высочайшую стабильность и безопасность, предпочитая долговременную поддержку непрестанной изменчивости (пример – изменение подсистемы виртуальной памяти посреди серии Linux 2.4).

Настольный вариант FreeBSD нередко отстает от Linux в поддержке новейшего оборудования, а Ports, система построения ПО, утратила уникальность после выхода Gentoo. Однако это, быть может, удобнее, чем море дистрибутивов Linux – одна кодовая база, один источник документации, одно централизованное хранилище ПО значительно упрощают жизнь пользователей FreeBSD. Ведь куда проще сказать «У меня FreeBSD 6.1», чем «У меня FooLinux на ядре 2.6.12, *Glibc 2.4.0* и *X.org 7.0.0*…»

ВЕРДИКТ LINUX FORMAT

ФУНКЦИОНАЛЬНОСТЬ	8/10
ПРОИЗВОДИТЕЛЬНОСТЬ	8/10
ПРОСТОТА ИСПОЛЬЗОВАНИЯ	7/10
ДОКУМЕНТАЦИЯ	10/10

Серьезную причину для перехода с 6.0 на 6.1 найти трудно, но FreeBSD снова нерушима, как скала.

РЕЙТИНГ 8/10

СИСТЕМА РЕЗЕРВНОГО КОПИРОВАНИЯ

Amanda 2.5

НА ДИСКЕ

На странице слева – *Amanda*, выдающаяся свободная система резервного копирования. **Грэм Моррисон** решил разобраться, так ли уж она хороша.

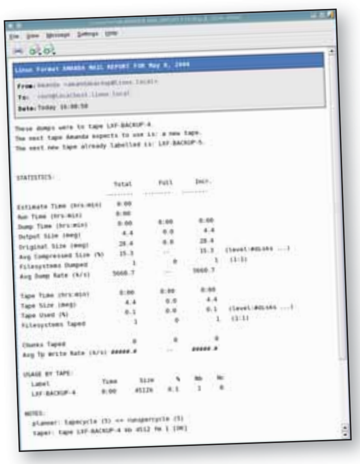
САМОЕ ГЛАВНОЕ

Резервирует данные от Linux-, Solaris- и Windows-клиентов на центральном сервере. Аналоги: *NetVault* (страница напротив) или *Arkeia Smart Backup* (*LXF78*).

- **РАЗРАБОТЧИК:** Университет Мериленда
- **САЙТ:** www.amanda.org
- **ЦЕНА:** Бесплатно по лицензии GPL

Взгляните на экранный снимок справа; теперь – на титульный лист. Знаем, о чем вы подумали, и вы совершенно правы: интерфейс *Amanda* красотой не блещет. В противоположность лощеному интерфейсу *NetVault*, внешность *Amanda* непритязательна: GUI отсутствует, и никуда не денешься от командной строки. Даже при установке *Amanda* в виде RPM не обойтись без некоторой возни с файлами конфигурации (потребуется известный опыт). Зато, как только все «на мази», в командной строке нужды уже нет – ну, разве что случится катастрофа…

Что необычно для открытого проекта – разработчики позаботились о вариантах RPM не только для SUSE и Fedora, но и для корпоративных версий обоих дистрибутивов. *Amanda* разработана в Университете Мериленда, отсюда и название – Advanced Maryland Automatic Network Disk Archiver (Продвинутый Мерилендский Автоматический Сетевой Дисковый Архиватор).



Результаты каждого копирования, включая отчет о занятой памяти, высылаются на на e-mail администратора.

Как и *NetVault*, *Amanda* работает по принципу «клиент-сервер». Сервер – ленточный хост (tape host), выдающий подключения ленточным же клиентам (tape client). В случае необходимости *Amanda* может работать с сотнями клиентов, причем не только с другими Linux- и Solaris-машинами, но и с компьютерами Windows (через Samba).

Нетрудно догадаться, что для резервного копирования ленточному хосту нужны ленточные накопители, но если у вас под Linux уже работает такой накопитель, то с *Amanda* он уживется почти наверняка. *Amanda* пользуется собственными протоколами для клиент-серверного соединения, но работает на стандартном Linux-оборудовании; правда, для него необходимо будет составить определение. Этот процесс способен занять от пяти до восьми часов; в итоге *Amanda* определит доступный объем хранилища и среднюю скорость передачи (хотя и в собственной базе данных *Amanda* заложены определения порядка 14 наиболее распространенных типов ленточных накопителей, среди них – обычные устройства от Quantum, Sony и HP). Если ленточный накопитель для вас избыточен, с помощью *Amanda* можно создать виртуальный ленточный накопитель, а затем вписать его объем в DVD или CD. Значительное преимущество нынешней версии состоит в поддержке многотомной резервной копии – системному администратору больше не нужно всякий раз соизмерять объем сохраняемых данных с размерами носителя.

Имеется поддержка смены лент, но чтобы ей воспользоваться, понадобится известное умение: в *Amanda* включен лишь простейший скрипт, который почти наверняка придется править.

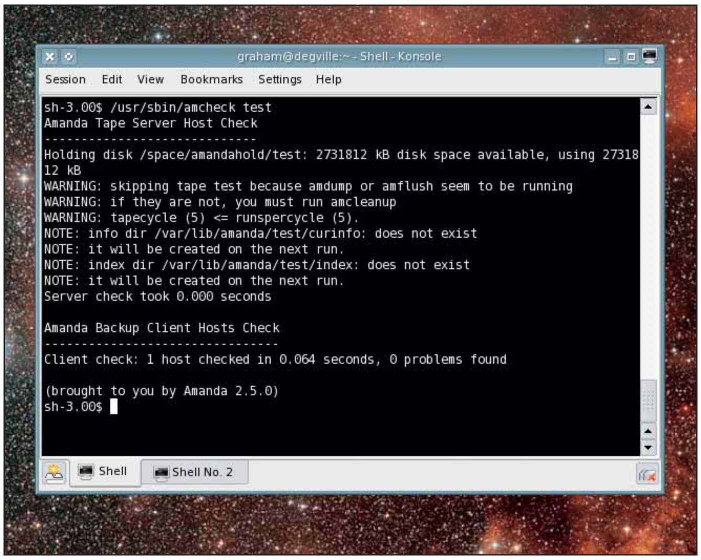
Тонкая работа

Один из важнейших критериев оценки любой программы резервного копирования – возможная степень сжатия данных. В *Amanda* и это под контролем, но кое-что зависит от инсталлированного ПО. Для большинства пользователей хорош *Gzip*, дающий даже большую степень сжатия, чем аппаратное (обычно около 2:1). Есть выбор между сжатием на стороне клиента (уменьшает нагрузку на сеть за счет использования клиентских процессоров), сжатием на стороне сервера (облегчается работа клиентов) и аппаратным сжатием. Новшество настоящей версии – возможность назначения политики в зависимости от типов файлов, это сохраняет уйму времени на попытках сжатия и без того сжатых файлов (некоторые аудио- и фото-форматы).

Безопасность – еще одно важное достоинство программы. Безопасность авторизации и передачи данных между клиентом и сервером усилена поддержкой Kerberos 4/5 и OpenSSH. Последний включается простым изменением строк 'auth' главного файла конфигурации, как на серверной, так и на клиентской стороне. Если *Amanda* найдет ключи SSH по умолчанию, то ими и воспользуется, но есть возможность прямого ввода таких ключей в файле конфигурации клиента.

Как вы уже заметили, мы часто упоминали о необходимости правки файлов конфигурации *Amanda* – зато других проблем просто нет. Это инструмент системного администратора, а не игрушка для взрослых. Вот почему на сайте *Amanda* представлен столь внушительный список фирм и консультантов, всегда готовых помочь в настройке – или, по их выражению, «внедрению». Существует также корпоративная версия от Zmanda, для платных подписчиков.

Решение недурное. Даже если в вашей системе все уже отлажено, уверенность в полной сохранности критически важных данных очень украшает жизнь. *Amanda* – необычайно полезный инструмент в умелых руках; и если ваши руки таковы, приложите их к делу и исключите малейший риск потерь! LXF



Смотреть не на что… *Amanda* – лабиринт файлов конфигурации и серия командных строк, оттого и картинка скучная.

ВЕРДИКТ LINUX FORMAT

ФУНКЦИОНАЛЬНОСТЬ	9/10
ПРОИЗВОДИТЕЛЬНОСТЬ	7/10
ПРОСТОТА ИСПОЛЬЗОВАНИЯ	4/10
ДОКУМЕНТАЦИЯ	7/10

Трудна в настройке, зато свободна и превосходно оснащена

РЕЙТИНГ 7/10

СИСТЕМА РЕЗЕРВНОГО КОПИРОВАНИЯ

BakBone NetVault 7.4

…а справа – необычная для нашего издания, весьма недешевая вещь. *NetVault* стоит кучу денег и может хранить терабайты информации, но **Грэм Моррисон** сомневается, что этого хватит для победы над чарами свободной *Amanda*.

САМОЕ ГЛАВНОЕ

Клиент-серверная программа резервного копирования с графическим интерфейсом. Аналоги: *Amanda 2.5* или *Arkeia Backup*.

- **РАЗРАБОТЧИК:** BakBone Software
- **САЙТ:** www.bakbone.com
- **ЦЕНА:** $1300+НДС (пять клиентов с 12-слотовым переключателем накопителей и круглосуточная поддержка 24/7)

NetVault – ещё один тяжеловес среди инструментов резервного копирования, но отнюдь не свободный. Он пользуется собственными протоколами и поддерживает гораздо более узкий диапазон устройств, чем *Amanda*, зато прекрасно оснащён. Если вас не запугает 600-страничное руководство администратора, то уж 23 вкладки панели конфигурации запугают непременно; и все-таки инсталлировать *NetVault* вовсе не трудно, особенно в сравнении с *Amanda* 2.5. Единственный скрипт позволяет установить любую клиент-серверную комбинацию или просто клиента.

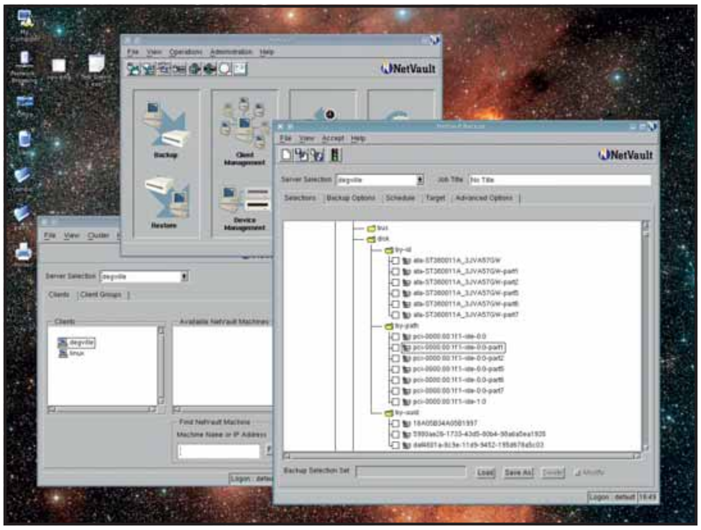
Системному администратору *NetVault* в основном придётся работать с двумя приложениями. Первое – вышеупомянутая 23-страничная конфигурационная панель, именно отсюда настраиваются все низкоуровневые параметры *NetVault*: конфигурация брандмауэра, установка и удаление модулей, аудит и лицензирование. Второе – *GUI NetVault*, известный в командной строке под именем *nvgui*. Это приложение одинаково как для клиента, так и для сервера, и хотя оно отделено от главной конфигурационной панели, именно здесь происходит каждодневная работа по добавлению новых устройств и подключению клиентов, а также просмотр журналов системы.

После инсталляции первым делом надо добавить клиентов *NetVault* к серверу. Когда программа установлена у клиента, достаточно будет просканировать сеть из Панели управления клиентами (Client Management panel) на сервере – появятся все доступные клиенты, и для включения в список на резервное копирование нужно будет просто щёлкнуть по ним мышью. Графический интерфейс скорее функционален, чем красив, и больше похож на старые приложения Motif, чем на новейший инструмент. Имеются также некоторые шероховатости: возможно открытие нескольких одинаковых окон сразу – такого мы не наблюдали с конца 90-х.

Опознав клиентов, можно настраивать носители архива. Выбор в *NetVault* беднее, чем в *Amanda*, но и здесь налицо десятки ленточных накопителей и NAS (network-attached storage, сетевых хранилищ), поэтому проблемы вряд ли будут. Как и в *Amanda*, возможно сохранение в локальной файловой системе, с помощью 'virtual library' – «виртуальной библиотеки», по терминологии *NetVault*; можно даже назначить целый диск в качестве самостоятельного виртуального устройства. Настроив клиентов и носители информации, смело приступайте к архивированию данных.

Иметь или не иметь?

NetVault почти везде использует систему модулей – то есть вы платите только за



Функциональный интерфейс пользователя позволяет просматривать папки, файлы и даже устройства. В список архивации они добавляются одним щелчком мыши.

действительно необходимые вещи, большинство же пользователей вполне устраивают умолчания. Единственное исключение – для случая, когда необходимо копировать базу данных без остановки её работы, тут уж понадобится доплата; имеются модули для работы с *Oracle*, *PostgreSQL* и *MySQL*.

Важнейший из стандартных модулей служит для прохода по клиентским файловым системам. Это больше, чем просто навигация по файлам и каталогам клиентской машины: существуют специализированные механизмы для каждого типа ФС. Так, если вы занялись файловой системой Linux-клиента, модуль знает, как обращаться с особыми файлами из **/proc** или **/dev**. Есть даже доступ к устройствам в **/dev** и просмотр оборудования по идентификатору – это удобно при архивировании диска целиком. Мы уже намекали, что другие файловые системы поддерживаются тоже – значит, доступны и Windows-клиенты. С помощью того же модуля вы получаете доступ и к реестру Windows, и к базе данных классов COM+. Конечно, с файлами и папками можно работать и обычным способом.

По умолчанию, резервное копирование делается инкрементально – иными словами, *NetVault* сохраняет лишь разницу между последней и текущей копией. Для полного восстановления данных на диске нужно восстановить самую первую копию, затем каждую успешную из серии последующих инкрементальных. Эта задача решается превосходным модулем, избавляющим вас от мороки ручного восстановления.

NetVault превосходно «упакован», и вы всегда чувствуете себя уверенно. Настройка конфигурации и управление копированием с помощью консоли и GUI – это две большие разницы; правда, не столько в удобстве, сколько в наглядности. Гораздо легче понять суть своих действий, когда вы видите свой маршрут по файловой системе клиента, а задания добавляются в очередь у вас на глазах. Опытного администратора этой косметикой, конечно, не проймёшь, но по крайней мере, в *NetVault* вы своими глазами видите, куда ушли ваши деньги. **LXF**

ПОЧЕМУ ИМЕННО ЛЕНТА?

Несмотря на DVD и винчестеры, ничего лучше ленты для резервных копий пока не придумано.

Не думайте, что магнитофоны упоминаются лишь в исторических трактатах: лента и резервная копия – до сих пор синонимы. Хранить данные на ленте значительно дешевле, чем на любом другом носителе, а содержать сотни гигабайт информации на одной магнитофонной катушке удобнее, чем на десятках DVD. Многие программы Linux специально созданы для прямой работы с лентой, лучший пример – tar. Он хорошо знаком всем нам как распаковщик файлов из Интернета, но мало кто знает, что tar – сокращение от Tape Archiver (Ленточный Архиватор). Старые ленточные накопители могли записывать данные блоками по 512 байт, отсюда и внутренний стандарт для tar. Amanda 2.5 использует tar для сжатия и конкатенации файлов в резервной копии, однако вместимость ограничивается объёмом одной ленты. NetVault в этом отношении продвинулся значительно дальше: он может использовать автоматические переключатели устройств – ёмкость хранилища становится почти неограниченной.

ВЕРДИКТ LINUX FORMAT

ФУНКЦИОНАЛЬНОСТЬ	9/10
ПРОИЗВОДИТЕЛЬНОСТЬ	8/10
ПРОСТОТА ИСПОЛЬЗОВАНИЯ	7/10
ОПРАВДАННОСТЬ ЦЕНЫ	6/10

Дорого, но по средствам тем, у кого размер жесткого диска соизмерим с размерами бюджета.

РЕЙТИНГ 8/10

Distrowatch

Ежемесячная сводка новостей дистрибутивов Linux.

ВОСХОД SUSE



Ладислав Боднар основатель, руководитель и сотрудник DistroWatch.com.

Когда Novell приобрёл SUSE в ноябре 2003-го года, пользователи забеспокоились о будущем любимого дистрибутива. Каковы истинные намерения Novell? А вдруг сетевой гигант превратит SUSE в эксклюзивную операционную систему для крупных корпораций и проигнорирует домашних пользователей? Или, чего доброго, прекратит работу над дешёвым коробочным продуктом и сфокусируется на гораздо более прибыльном серверном рынке?

> «ОТКРЫТО МНОГОЕ ИЗ КОДА SUSE И КОДА САМОЙ ФИРМЫ NOVELL.»

К счастью, теперь мы знаем, что этот пессимизм был необоснован. SUSE Linux по-прежнему продаётся в Европе и Северной Америке, причём Novell, для привлечения сторонних разработчиков, вообще открыла дистрибутив и предоставила к нему свободный доступ, как только вышла стабильная версия. Лучше того – большая часть кода SUSE и собственного кода Novell переведена под лицензию GPL: приложения *Yast*, *AppArmor* и, совсем недавно, *Xgl* и *Compiz*.

Пользователи Linux могут радоваться; однако инвесторы Novell скорее всего не так довольны. Компания пока не сумела превратить успех SUSE Linux в источник дохода, и если ситуация вскоре не изменится, Novell окажется под давлением своих акционеров. Но есть и хорошие новости – Novell в ближайшие месяцы собирается выпустить два важных продукта: SUSE Linux Enterprise Server и SUSE Linux Enterprise Desktop, которые должны принести прибыль, достаточную для всеобщего счастья. Будем надеяться, что эта стратегия принесёт успех.

Неоконченная Симфония

Symphony OS – новый игрок на рынке настольных систем – достигла бета-статуса.

После установки нескольких самых распространённых дистрибутивов легко заключить, что вы видели их все. Но Райан Квинн [Ryan Quinn], создатель Symphony OS, знает непохожий.

На данный момент, Symphony OS – без сомнения, одна из самых необычных настольных ОС. Вместо копирования традиционного способа доступа к приложениям и утилитам через систему каскадных меню эта система использует так называемые десклеты – наборы многофункциональных утилит для запуска программ, подключения к Интернет, поиска файлов и предоставления всевозможной полезной информации. Десклеты специально спроектированы для предоставления простого доступа ко всему, что может понадобиться большинству пользователей.



***Mezzo*, рабочий стол Symphony OS, легко конфигурируется через Desktop Manager. Видите минимизированную иконку *Gimp*?**

Фундаментальную часть Symphony OS образует рабочий стол, носящий имя *Mezzo*. Помимо вышеупомянутых десклетов, *Mezzo* отличается от большинства традиционных рабочих столов тем, что у него нет панели задач: если приложение минимизировано или потеряло фокус, оно становится либо иконкой, либо миниатюрным окном в нижней части экрана. *Mezzo* также предоставляет четыре других виртуальных рабочих стола с добавочными десклетами. Все они доступны по щелчку на иконки, расположенные в четырёх углах экрана. Десклеты можно настраивать, и по мере взросления проекта, скорее всего, созданные сообществом десклеты покроют все мыслимые виды задач – подобно расширениям *Firefox*.



Создатель системы Райан Квинн – Perl-хакер, интересующийся дизайном GUI.

Хотя Symphony OS может использовать обычный набор утилит Debian для установки программного обеспечения (система построена на базе нестабильной ветки Debian), разработчики создали графическое приложение для установки дополнительных программ и удаления ненужных пакетов. *One-Click Software* (так называется это приложение) представляет собой web-сервис, предоставляющий доступ к определённому количеству категорий программ, предназначенных для установки одним щелчком, включая игры, Интернет, мультимедиа и офисные приложения.

Непременно попробуйте эту систему в действии, и хотя её последняя версия (на момент написания статьи это 2006-05 beta) пока что не является полнофункциональной и содержит изрядное количество очевидных ошибок, она гарантирует вам несколько часов интересного времяпрепровождения. И даже если позже вы отбросите идею перехода на *Mezzo*, вы, наверное, согласитесь, что всем надо встать и поаплодировать разработчикам Symphony OS за смелость мыслить по-другому.

www.symphonyos.com

ЗНАМЕНИТОСТИ MEZZO

Идея новаторского рабочего стола *Mezzo* пришла в голову Джейсону Списаку, он в данный момент является лидером проекта Symphony наравне с Райаном Квинном. Списак также – сооснователь Lycoris и по совместительству актёр, озвучивающий мультфильмы и игры – он принимал участие в озвучивании сериала 2001-го года Transformers: *Robots in Disguise*.

Разбиение – запросто!

GParted Live CD – бесплатное приложение типа Live CD для разбиения жёсткого диска

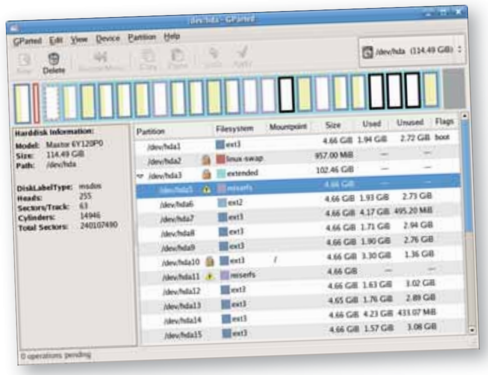
GParted Live CD – не столько дистрибутив Linux в привычном смысле этого слова, сколько утилита, предназначенная для упрощения процедуры разбиения жёсткого диска. Эту программу можно считать аналогом *Partition Magic* в мире бесплатных программ; и действительно, Live CD размером 31 МБ – не только великолепный инструмент разбиения диска, но он также находится в свободном доступе, его можно бесплатно использовать, модифицировать и распространять.

Как следует из его имени, GParted Live CD – это дистрибутив, загружающийся и работающий с компакт-диска. После парытройки вопросов о выборе языка и раскладки клавиатуры, загружается *GParted* – графический интерфейс к *GNU Parted* с поддержкой drag-and-drop. Он сканирует жёсткие диски, определяя их текущее разбиение.

GParted поддерживает большинство популярных файловых систем: Ext2, Ext3, FAT16, FAT32, JFS, ReiserFS, Reiser4, NTFS, XFS и несколько других. Все эти файловые системы могут быть созданы с помощью нескольких щелчков мыши, но изменение их размеров – задача более сложная: некоторые системы, например, ReiserFS и Reiser4, не могут менять свой размер, а большинство типов файловых систем (кроме FAT16 и FAT32) нельзя перемещать. Главное, что уменьшение раздела NTFS (системы, используемой в Windows XP) поддерживается хорошо!

Помимо очевидной способности работать с разделами, GParted также обучен копировать разделы и жёсткие диски, что делает его отличной утилитой резервного копирования и восстановления.

gparted.sourceforge.net



***GParted*: и утилита разбиения жёсткого диска, и информатор о файловых системах.**

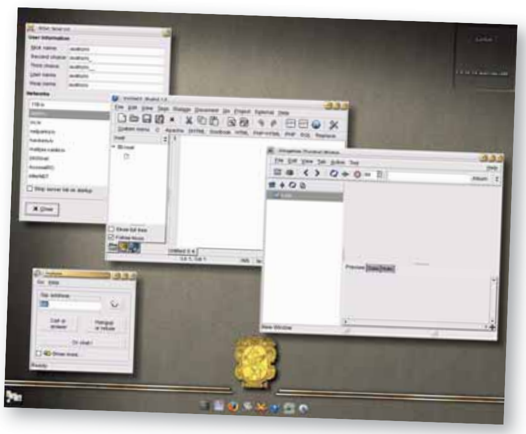
Мал, но мил

Austrumi Live CD – лёгкий, быстрый и эффектный.

При попытке ранжировать программные проекты по степени близости к компьютерному эквиваленту понятия нирваны Austrumi Live CD подберётся к самой вершине списка. А как иначе описать систему, которая умещается на мини-CD размером 50 МБ, работает с нечеловеческой скоростью, отлично выглядит и включает столько программ, что с трудом верится в её загрузку всего за несколько минут? Ну, а если вам и этого мало – она ещё и понимает по-латышски!

Созданный Андреем Мейнертсом [Andrej Meinerts], этот дистрибутив – настоящее чудо среди других мини-дистрибутивов Linux. Он загружается, целиком копируя себя в ОЗУ для более быстрого доступа к приложениям и для высвобождения CD-привода под другие, более интересные задачи, вроде проигрывания видео. Он также сам опознаёт состав аппаратуры, и не успеете вы опомниться, как окажетесь в *Enlightenment 17* – нестабильной версии прекрасного легковесного рабочего стола, славящегося необычными визуальными эффектами.

Крошечный Austrumi включает в себя почти все мыслимые типы приложений рабочего стола. Тут есть *AbiWord* и *Gnumeric* для офисных задач, *Bluefish* для редактирования HTML, *Gimp* и *Inkscape* для редактирования графики, *Firefox*, *XChat*, *GFTP* и *Linphone* для Интернет, *MPlayer* для проигрывания мультимедиа, несколько игр, утилиты конфигурирования и даже серверы. Поскольку все эти программы уже загружены в память, запуск любой из них не займёт больше пары секунд.



Рабочий стол *Austrumi* – *Enlightenment* – куда красивее, чем *Fluxbox* из Damn Small Linux.

Почему бы и вам не скачать последнюю версию? Но учтите, как только она окажется на вашем компьютере, будет очень сложно вернуться к старому дистрибутиву. LXF

http://sourceforge.net/projects/austrumi

ХИТ-ПАРАД ДИСТРИБУТИВОВ

10 самых посещаемых страниц на DistroWatch.com в мае

(среднее число визитов в день)

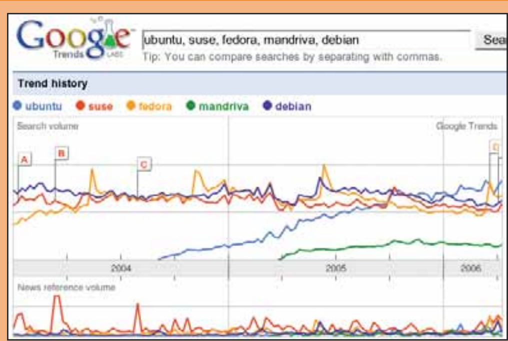
1	Ubuntu Linux	2,817
2	SUSE Linux	2,596
3	Fedora Core	1,259
4	Mepis Linux	1,047
5	Mandriva Linux	914
6	Damn Small Linux	902
7	Debian GNU/Linux	745
8	FreeBSD	735
9	PCLinux OS	646
10	Knoppix	624

DistroWatch.com отслеживает популярность дистрибутивов, основываясь на количестве посещений сайтов, посвящённых конкретным дистрибутивам. Хотя эти цифры и не отражают настоящее количество инсталляций, они являются индикатором популярности дистрибутива на данный момент времени.

ИНТЕРЕС К UBUNTU РАСТЁТ И РАСТЁТ

В Google за последнее время создано много полезных инструментов; один из них – Google Trends (**www.google.com/trends**), бесплатный сервис, позволяющий отследить тенденции поисков по ключевым словам на этой популярной поисковой машине. Какие темы интересуют людей и как этот интерес менялся с течением времени? Google Trends всё это знает.

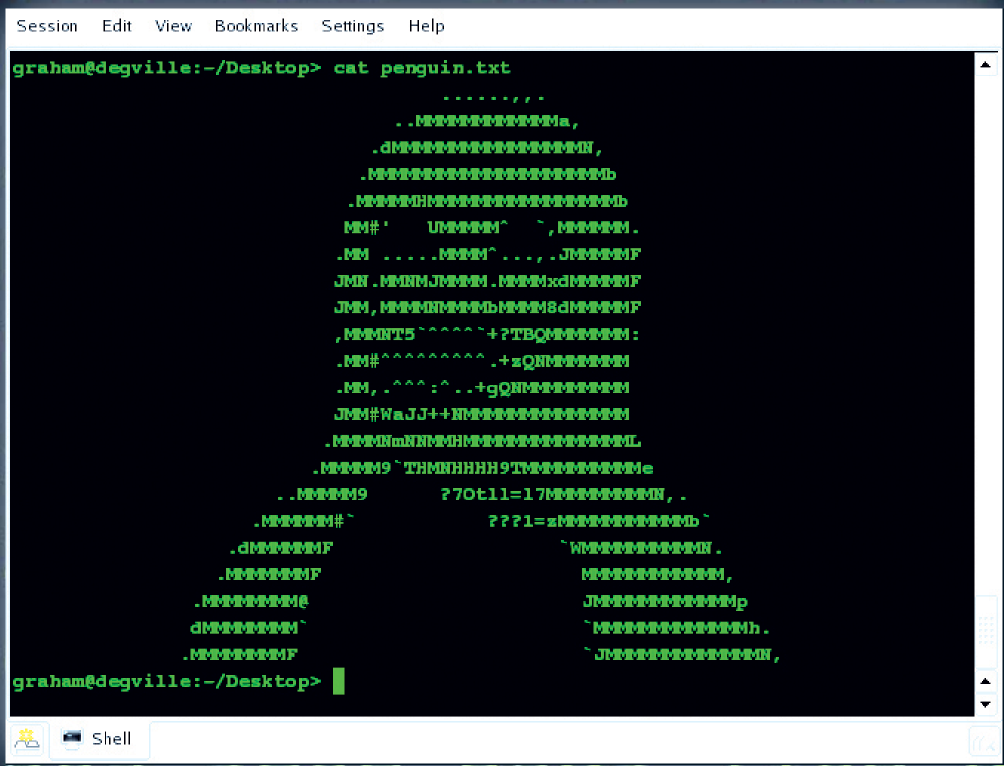
Мы приводим графики количества запросов на пять слов: Ubuntu, SUSE, Fedora, Mandriva и Debian. Результат подтверждает феноменальный рост популярности Ubuntu – кривая начинается с нулевого числа поисков в середине 2004-го и в итоге опережает все прочие поисковые слова в мире Linux. Тенденции поиска по остальным большим дистрибутивам остаются на постоянном уровне, с пиками около моментов релизов. Исключение составляет Mandriva, к ней интерес за последний год был очень ровным.



Были уже и Google Maps и Google Fight, а теперь Google Trends сообщает, что нынче в моде.

Сравнение

Каждый месяц мы просматривает тонны ПО, чтобы вам не приходилось заниматься этим самостоятельно.



X-ТЕРМИНАЛЫ

НА ДИСКЕ

Ах, настоящая работа с компьютером! **Нейл Ботвик** ныряет в омут терминалов, предлагаемых Linux.

Большинство пользователей Linux начинают с использования графического интерфейса. Основные дистрибутивы снабжены отличными инсталляторами и инструментами конфигурации, с которыми вы могли прохлаждаться в комфорте графического интерфейса пользователя (GUI), но рано или поздно каждый из нас покидал его ради командной строки. К примеру, какое-то нужное вам приложение не поставляется в качестве пакета в вашем дистрибутиве и поэтому приходится устанавливать его из исходных текстов. Или оказывается, что единственные доступные инструкции для выполнения вашей задачи подразумевают использование оболочки (shell).

А еще уважительнее следующая причина: вы уж так наслушались, что оболочка – это хорошо, что захотели сами в ней поработать. Можете нажать Ctrl+Alt+F1 и выйти в чистую консоль Linux, но зачастую куда удобнее запустить сессию командной строки в окне на вашем рабочем столе – здесь-то и появляется *X*-терминал, или, полностью, эмулятор *X*-терминала (название выдает, что воспроизводится тупой старый терминал ПК прошлых лет).

Запуск терминала на рабочем столе означает, что вы сможете прочитать инструкции в графическом web-браузере или почтовой программе, перед тем как выполнить их в терминале. Все терминалы, рассмотренные здесь, позволяют также вставлять скопированный из другого окна текст, так что вам, возможно, даже не придется набирать его – только не вздумайте по требованию какой-нибудь web-странички вставить **rm -fr /*** !

Проверка скорости

Считается, что *X*-терминалы обладают лишь базовыми функциями, но в большинстве своем они имеют широкий выбор пользовательских опций или интегрированы с определенной средой рабочего стола. Выбор эмулятора терминала – процесс глубоко личный; здесь мы постарались вам дать представление о том, какие есть приложения и каковы их сравнительные достоинства и недостатки. Терминалы тестировались в разных средах, в основном это был Athlon 64, с KDE под *Gentoo*, и *iBook* с *Gentoo* и *Fluxbox*.

Чтобы вам помочь, мы сконцентрировались на разнице между ними. При этом рассматривались: количество имеющихся функций, простота использования и модификации этих функций, влияние внешних украшений – например, прозрачности – на производительность терминала и программы: увеличивается она или, наоборот, снижается.

Производительность – у командной оболочки? А почему бы и нет: разница в скорости, с которой эти программы выводят текст, огромна; на большинство операций это, может быть, и не повлияет, но иногда бывает важно. Мы измеряли скорость с помощью программы, которая выводила длинный текст в терминал и измеряла затраченное на это время. Если ваше отношение к Linux серьезно (а раз уж вы читаете *Linux Format*, так оно, видимо, и есть), вы проводите немало времени в своем любимом терминале, так что читайте внимательно…

НАШ ВЫБОР ДЛЯ СРАВНЕНИЯ

Yakuake

Quake знакомится с командной строкой

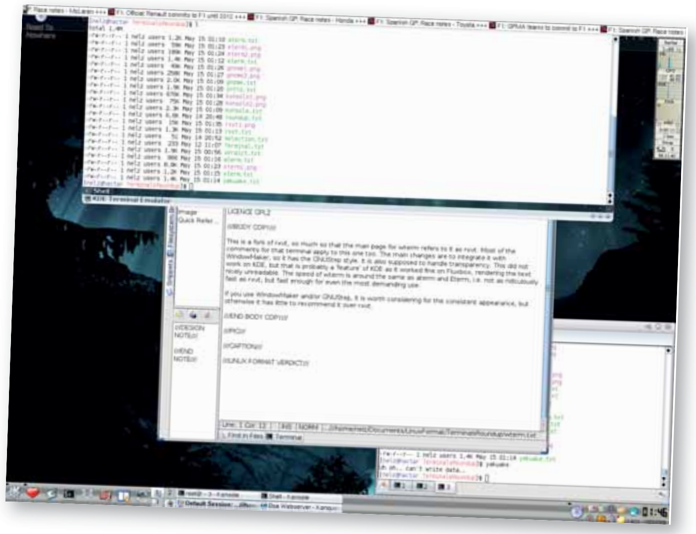
- **ВЕРСИЯ:** 2.7.5 • **WEB:** http://extragear.kde.org/apps/yakuake
- **ЦЕНА:** Бесплатно по лицензии GPL

***Yakuake* – это ответвление Kuake,** заброшенного больше двух лет назад. Это терминал в стиле *Quake*, вываливающийся из верхней части дисплея, когда вы нажимаете клавишу. Если, по-вашему, эмулятор терминала на базе одной из функций стрелялки – это круто, берите его. Терминал *Quake* основан на предположении, что может потребоваться быстро показать и спрятать его в любой момент игры; такой постулат можно применить и к использованию обычного рабочего стола – если вам надо выполнить пару командных строк, то не придется открывать терминал, выполнять команды и закрывать его снова: просто нажмите горячую клавишу – и терминал перед вами.

Yakuake сохраняет свое состояние между сессиями, и это куда полезнее, чем открытие нового окна всякий раз, как вам понадобится командная строка. Это ставит его где-то между постоянно открытым окном *X*-терминала и мини-консолью KDE, которая позволяет вам запускать команды, но не дает увидеть результат.

Yakuake основан на Konsole, и позаимствовал оттуда функции типа полос прокрутки, настраиваемых вкладок и прозрачности. *Yakuake* даже медленнее, чем Konsole (см. стр. 35), но это не тот терминал, который стоит использовать для задач с большой нагрузкой, требующих скорости.

Еще одно различие между *Yakuake* и окном *X*-терминала – *Yakuake* открывается на всех рабочих столах, поверх всех окон, благодаря чему его не потеряешь. Это отличное свойство в следующем релизе будет опциональным. Вы настраиваете свойства Konsole обычным способом, а собственные настройки *Yakuake* можно изменить, редактируя недокументированный файл конфигурации на **~/.kde/share/config/Yakuakerc**. Полезная информация в этом файле позволит вам сэкономить немало времени.



Yakuake – **нетривиальная альтернатива стандартам Gnome и KDE, и его стоит исследовать.**

Yakuake может изменять внешний вид, и хотя поставляется только с одной «шкуркой», остальные можно найти на www.kde-look.org, кроме того, там имеется подсказка по созданию собственного дизайна.

ВЕРДИКТ LINUX FORMAT

Этот терминал скорее годится не для напряженных сессий оболочки, а для быстрых задач, но уж для них-то он весьма полезен.

РЕЙТИНГ 7/10

Yeahconsole

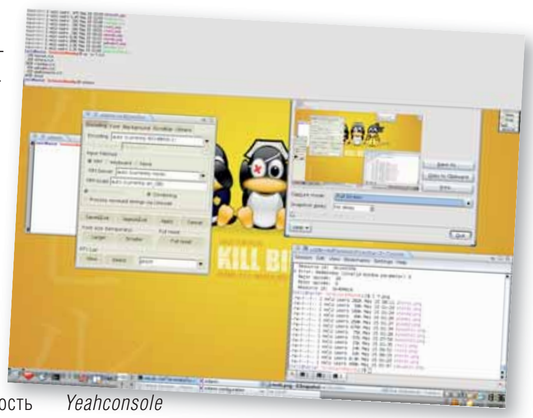
Всплывающий терминал в стиле *Quake*

- **ВЕРСИЯ:** 0.3.4 • **WEB:** http://phrat.de/yeahtools.html
- **ЦЕНА:** Бесплатно по лицензии GPL

***Yeahconsole* – еще один терминал в** стиле *Quake*, похожий на *Yakuake*. Различие между ними в том, что *Yakuake* основан на Konsole, и поэтому для работы ему нужна хотя бы часть KDE, тогда как *Yeahconsole* использует свободный от зависимостей Xterm.

Если вы нажмете нужное сочетание горячих клавиш, в верхней части экрана выплывет окно *Yeahconsole*. Оно появится на всех ваших виртуальных рабочих столах, поверх всех окон. У вас все время будет одна и та же сессия терминала – он просто либо прячется, либо показан, а не остановлен и перезапущен. Можно, правда, настроить его и на старт сессии с нуля при каждом появлении, но это не является установкой по умолчанию и для большинства пользователей не особо удобно.

Yeahconsole настраивается через файл **~/.Xdefaults**; можно задать размеры и положение окна, скорость всплывания окна и горячие клавиши. Некоторый набор терминалов можно использовать и внутри *Yeahconsole*. По умолчанию – *Xterm*, снабженный лишь самым необходимым, но зато всегда имеющийся под рукой. Можно также взять терминалы *Urxvt* или *Urxvtc* из ответвления *Rxvt* с добавкой Unicode.



Yeahconsole **предлагает мгновенный доступ к терминалу с оттенком легкой** *Quake*-**ностальгии.**

Производительность *Yeahconsole* отнюдь не выдающаяся, особенно при использовании *Xterm*, но дело не в этом. *Yeahconsole* предназначен не для трудных терминальных задач, а для быстрого выполнения случайных команд, не сходя с рабочего стола, и с этой задачей справляется очень хорошо. Он не обладает богатством функций *Yakuake*, но станет хорошим дополнением рабочего стола для всех, кто не работает в KDE.

ВЕРДИКТ LINUX FORMAT

Далек от гибкости *Yakuake*, но зато полегче и не зависит от рабочего стола.

РЕЙТИНГ 6/10

ТЕРМИНАЛЫ, ОБОЛОЧКИ И КОМАНДНЫЕ СТРОКИ

При описании среды командной строки на терминалах используется целый ряд терминов (извините за каламбур), порой взаимозаменяемых. Мы попытаемся выловить рыбку смысла в этой мутной водице.

Использование слова «терминал» восходит к прошлому, ко временам больших компьютеров, доступ к которым предоставлялся через отдельное устройство, называемое тупым терминалом (dumb terminal). Эмулятор X-терминала, рассматриваемый здесь – это программа, выполняющая ту же функцию, но в окне, расположенном на рабочем столе. Обычно терминал используется для передачи команд компьютеру, на котором он запущен, но можно соединить его окно и с другим компьютером, через сеть.

Консоль, или виртуальная консоль – это чисто текстовый интерфейс. Именно ее вы видите, когда загружаете Linux без использования *X*: возникающая подсказка ввести учетную запись находится на виртуальной консоли. Большинство дистрибутивов Linux имеют шесть консолей, в которые можно выйти, нажав Ctrl+Alt+F1…6 (Ctrl+Alt+F7 обычно нажимают, чтобы выйти на рабочий стол).

Оболочка – это программа, работающая в терминале или виртуальной консоли. Оболочка – настоящий интерпретатор командной строки, который считывает набираемые вами команды и выполняет их. Существует несколько оболочек, но почти все дистрибутивы по умолчанию используют *Bash* (*Bourne Again Shell* – «*Возрожденная*» оболочка, намек на первичную оболочку *Bourne* в Unix), хотя многие ругаются с компьютером через оболочку *zsh* с www.zsh.org.

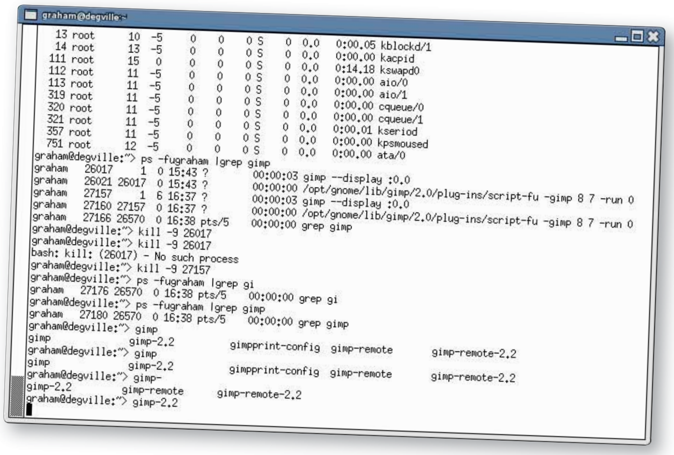
Xterm

Прототерминал: все еще скудный, все еще медленный

• **ВЕРСИЯ:** 212 • **WEB:** http://dickey.his.com/xterm
• **ЦЕНА:** Бесплатно по лицензии MIT X11

Это самый первый X-терминал, но он все еще в активной разработке. В нем не найдется ни вкладок, ни меню, ни прозрачности, ни профилей, и вообще никаких особых свойств, отличающих другие терминалы данного Сравнения. Вы можете подумать, что он зато быстрый и эффективный – а вот и нет. На самом деле *Xterm* медленнее абсолютно всех рассмотренных терминалов, причем здорово медленнее. Шрифты, цвета и эмуляцию, а заодно и другие опции, можно настраивать нажатием Ctrl и одной из трех кнопок мыши для вызова трех соответствующих меню (прочие установки указываются в командной строке), уж эти-то меню уникальны. Правда, они не позволяют сохранить текущую настройку – редактируйте файл **~/.Xresources**.

Кто-то, наверное, скажет, что *Xterm* не стоит выбрасывать на свалку, потому что благодаря отсутствию зависимостей он выручит в аварийной ситуации, но, если одна из наших машин дойдет до жизни такой, что на ней не заработает ни один из других описанных терминалов, мы уж лучше нажмем Ctrl+Alt+F1.



ВЕРДИКТ LINUX FORMAT

Причины рекомендовать этот терминал решительно отсутствуют – разве что смутное желание эмулировать что-нибудь этакое.

РЕЙТИНГ 3/10

Xterm самый медленный и бедный терминал нашего обзора – но зато зависимостей у него меньше всех.

Rxvt

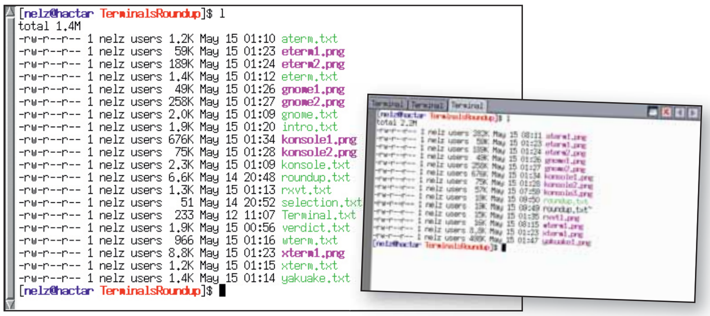
Функций – минимум, скорость – потрясающая

• **ВЕРСИЯ:** 2.7.10 • **WEB:** www.rxvt.org • **ЦЕНА:** Бесплатно по лицензии GPL

***Rxvt* («Наш виртуальный X-терминал»)** вознамерился стать быстрой и легкой заменой для *Xterm*, отказавшись от эмуляции Tektronix и меню, и достиг успеха: его скорость в 15 раз выше скорости *Xterm*. Достиг, однако, дорогой ценой: *Rxvt* лишь чуть получше простейшего терминала. Это вовсе не всегда плохо: базовые свойства плюс быстрота и отсутствие излишеств – многим пользователям именно это и нужно.

Настройка *Rxvt* не особо дружественная. Вся конфигурация устанавливается через параметры командной строки, а также добавлением опций в ваш файл **~/.Xresources**, и настройки нельзя менять, пока работает *Rxvt*. Единственная уступка любителям красивостей – возможность поместить фоновый рисунок в окне терминала. Некоторым это нравится, но по жизни приводит только к ухудшению читаемости текста.

Если вам нужна функциональность чуть пошире – попробуйте *Mrxvt*. По сути, это *Rxvt* со вкладками, при той же скорости и почти таком же минимуме функций.



Недостаток свойств _Rxvt_ искупает быстротой. Кому нужна версия со вкладками, переходите на _Mrxvt_ (не показан).

Вкладки – хорошее подспорье при работе более чем в одной сессии оболочки одновременно. Если вам из всех дополнительных функций, предоставляемых терминалами, нужна только эта, *Mrxvt* может оказаться лучшим выбором.

ВЕРДИКТ LINUX FORMAT

Rxvt имеет мало функций, использует мало ресурсов и очень быстр. Попробуйте его на старых системах.

РЕЙТИНГ 6/10

Wterm

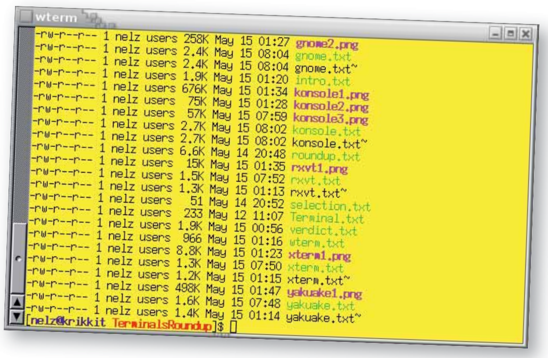
Оптимизирован для *WindowMaker* и *GNUstep*

• **ВЕРСИЯ:** 6.2.9 • **WEB:** www.wterm.org • **ЦЕНА:** Бесплатно по лицензии GPL

Это – ответвление *Rxvt*: man-страница *Wterm* ссылается на него как на *Rxvt*, и большинство ранних комментариев к *Rxvt* можно отнести и к *Wterm*. Основное различие – интеграция *Wterm* с *WindowMaker*, которая придает ему сходство с GNUStep.

Wterm также обладает настраиваемой прозрачностью, превращающей текст в нечто невразумительное – по крайней мере, при работе в *Fluxbox*. Это не работает в KDE, что, возможно, есть «не баг, а фича» (KDE отображает свой фоновый рисунок поверх корневого окна *X*). Поскольку *Wterm* не заботится об уступках хорошему вкусу, будьте осторожны при выборе цвета шрифта и фонового рисунка, если используете прозрачность, еще и потому, что великое множество цветов используется для отображения файлов разного типа. Терминал почти столь же быстр, как *Aterm* и *Eterm*, то есть не настолько неприлично быстр, как *Rxvt*, но его скорость удовлетворит и взыскательных пользователей.

Wterm гармонично сочетается с *WindowMaker* и/или GNUStep по внешнему виду; другие козыри против *Rxvt* у него вряд ли найдутся.



Будьте осторожны при выборе цветов для шрифтов и фона.

ВЕРДИКТ LINUX FORMAT

Если вам не нужна интеграция с *WindowMaker*, то от *Wterm* вам ждать нечего.

РЕЙТИНГ 5/10

Konsole

Прибамбасы-причиндалы… и звонки впридачу

• **ВЕРСИЯ:** 3.5.2 • **WEB:** www.kde.org • **ЦЕНА:** Бесплатно по лицензии GPL

***Konsole* — типичное приложение KDE:** в нем больше опций настройки, чем семечек в арбузе. Полосы прокрутки отображаются на любой из сторон окна либо не отображаются вообще. Вкладки могут появляться вверху, внизу или вообще нигде. Даже если они скрыты, *Konsole* умеет рулить несколькими терминалами в одном окне, а для переключения с одного на другой используется клавиатура — чтобы вывести на экран максимальное количество текста, войдите в полноэкранный режим, скрывающий вкладки и меню.

Konsole способен сохранять варианты настроек, включая текущую директорию и команды, которые вы хотите запустить, в профиле. Профиль текущей оболочки, в отличие от *Gnome Terminal*, сменить нельзя, но вы можете загрузить профиль в другую вкладку, удерживая кнопку слева внизу окна. Имеется несколько предопределенных профилей, включая работу в *Midnight Commander*.

Если у *Konsole* есть ахиллесова пята, то это – скорость (точнее, ее отсутствие). Возможно, для большинства пользователей терминалов скорость не играет важной роли, но в конфигурации по умолчанию *Konsole* работает действительно медленно. *Gnome Terminal* быстрее почти в два раза, а *Rxvt* прокручивает текст в восемь раз быстрее. Главным образом, в этом повинно использование в *Konsole* сглаженных шрифтов — запустите его с помощью опции **--noxft**, и скорость показа текста удвоится.

Сэкономить время

Если вы захотите войти в директории общего пользования, благодаря меню закладок *Konsole* (Bookmarks) понадобится меньше ручного набора; даже при наличии функции автодополнения те, кто медленно печатает, согласятся, что это меню — самый быстрый способ навигации. Поскольку *Konsole* — приложение KDE, почти каждое действие можно привязать к горячим клавишам (shortcut), которые вы сами назначаете.

Как и следовало ожидать, *Konsole* хорошо интегрирован с остальными программами KDE. Перетащите файл из *Konqueror* в *Konsole* – и у вас появится выбор: скопировать файл, или переместить его, или через cd забраться в его директорию. Причем это касается не только локальных файлов: для загрузки в текущую директорию *Konsole* вы можете перетащить даже ссылку с web-страницы. Документация у *Konsole* тоже хороша и отлично интегрирована с Информационным Центром KDE (KDE Help Centre). Есть и другие милые мелочи, например, функция уведомления о завершении команды в одной из вкладок, то есть мирно работая на одной вкладке, вы узнаете, что завершилось выполнение команды, запущенной на другой. Превосходно.



Терминал с невообразимым количеством функций. *Konsole* **сделает все, что вообще можно сделать – разве что порой слегка подтормаживая.**

> **ВЕРДИКТ LINUX FORMAT**
>
> *Konsole* – полнофункционален и всеобъемлющ, и делает работу в оболочке гораздо эффективнее. Но для увеличения скорости отключите сглаживание шрифтов.
>
> **РЕЙТИНГ 9/10**

Gnome Terminal

Богатый функциями терминал по умолчанию для Gnome

• **ВЕРСИЯ:** 2.14.4 • **WEB:** www.gnome.org • **ЦЕНА:** Бесплатно по лицензии GPL

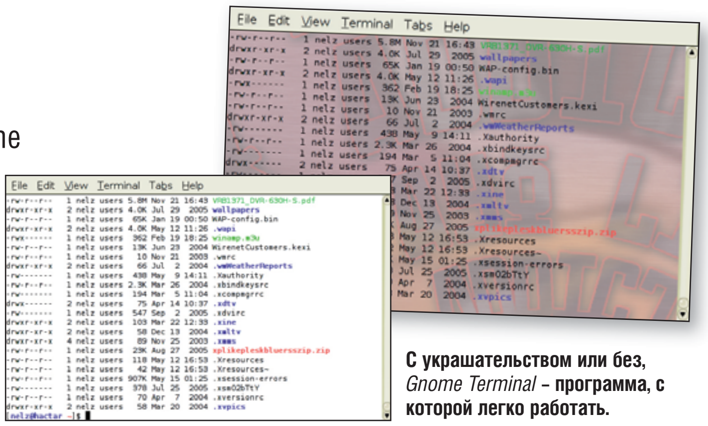
Если вы используете дистрибутив на основе Gnome, например, Fedora Core или Ubuntu, вы, наверное, и не знаете других терминалов. И это не так уж плохо: *Gnome Terminal* – программа добротная, с поддержкой таких функций, как вкладки или история прокрутки для помощи при работе в оболочке. Пусть над ними глумятся крутые пользователи терминалов, но зато они делают рабочий процесс более удобным и дружественным. Вкладки – тоже исключительно полезное дополнение, позволяющее открыть несколько оболочек или экранных сессий (screen sessions) в одном окне.

Gnome Terminal использует профили для сохранения свих настроек, так что переключение на совершенно непохожую конфигурацию сводится к выбору нового профиля из меню. Профиль содержит полный диапазон настроек, включая шрифты, цвета, положение полосы прокрутки (или ее отсутствие), цвет фона или прозрачность, а также работает ли оболочка в режиме «login shell». Вы можете также задать команду, которая будет запускаться вместо стандартной оболочки, что позволяет создавать профили для отдельных приложений, например, для *Midnight Commander* или *Lynx*.

Благодаря профилям настраивать *Gnome Terminal* проще, чем любой другой терминал из нашего **Сравнения**, но по умолчанию профили отсутствуют.

Gnome Terminal работает медленнее, чем большинство других терминалов в нашем **Сравнении**, тем не менее он почти в два раза быстрее Konsole (в режиме по умолчанию). Насколько эта быстрота важна для вас, зависит от того, нужно ли вам перемещать огромные тексты на медленном оборудовании: 12 000 строк в секунду на Athlon 64 – это намного больше, чем может прочесть большинство из нас.

Использование памяти зависит от среды рабочего стола, в которой вы работаете: этот терминал рассчитан на работу с Gnome, тогда все соответствующие библиотеки будут загружены и использование ресурсов будет минимальным. Кстати, у *Gnome Terminal* есть еще очень удобный справочник, так что вы сможете извлечь максимум пользы из имеющихся функций.



С украшательством или без, *Gnome Terminal* **– программа, с которой легко работать.**

> **ВЕРДИКТ LINUX FORMAT**
>
> Легко настраиваемый, хотя и обладающий меньшим количеством опций, чем *Konsole*, *Gnome Terminal* успешно дополняет рабочий стол Gnome.
>
> **РЕЙТИНГ 8/10**

Eterm

Терминал *Enlightenment* – быстр и привлекателен

• **ВЕРСИЯ:** 0.9.3 • **WEB:** www.eterm.org • **ЦЕНА:** Бесплатно по лицензии GPL

***Eterm* создан для интеграции со сре**дой рабочего стола *Enlightenment*, но работает с любым менеджером окон или с любой другой средой рабочего стола. По части скорости *Eterm* занимает место где-то посередине списка конкурсантов: вполовину медленнее, чем *Rxvt*, но намного быстрее, чем *Xterm* или *Konsole*. В отношении функциональности он тоже занимает промежуточную позицию, чего и следует ожидать от программы, разработанной для *Enlightenment*, привлекательного и легкого оконного менеджера.

Eterm имеет меню для настройки шрифтов, фона и прозрачности, так что нет необходимости лазить по справке и затем перезапускать терминал для того, чтобы все это изменить. Естественно, при желании вы можете также указать эти опции в командной строке или в конфигурационном файле.

На сайте **www.eterm.org/themes** много тем для *Eterm*. Скачайте тему, распакуйте в **$HOME/.Eterm/themes** – и при запуске *Eterm* сможете преобразить его внешний вид. *Eterm* отдает приоритет настройкам пользователя над настройками темы, что позволяет просто взять существующую тему, поковыряться в опциях и затем сохранить настройки пользователя, все из меню *Eterm*. *Eterm* быстр, мало весит и легко настраивается.



Eterm: **компромисс между минимализмом и перебором..**

ВЕРДИКТ LINUX FORMAT

Eterm находится ровнехонько посреди ассортимента – отличный выбор для тех, кто любит минимальные, но привлекательные среды рабочего стола.

РЕЙТИНГ 7/10

Aterm

Командная строка, доселе невиданная

• **ВЕРСИЯ:** 1.0.0 • **WEB:** http://aterm.sourceforge.net
• **ЦЕНА:** Бесплатно по лицензии GPL

Aterm является собой развитие *Rxvt*, с упором на приятные для глаза визуальные эффекты. Эффектов навалом: фоновые рисунки, прозрачность, полупрозрачность с настраиваемой насыщенностью тона (чтобы текст оставался читаемым). Можно даже обесцветить текст, щелкнув вне окна – очень удобно, когда вы хотите скрыть от своего босса, что не трудитесь в поте лица, а занимаетесь хакерством.

Чтобы воспользоваться некоторыми из этих функций, понадобится установить оконный менеджер *AfterStep*. Выберете вы эту программу или нет, будет зависеть от того, насколько для вас важна внешняя привлекательность: если вам нравится простой читаемый текст на простом контрастном фоне, то прозрачность и фоновые рисунки только увеличивают размер и снижают скорость работы программы, хотя *Aterm* перелопачивает тексты большого объема во вполне приличном темпе.



При всем блеске, *Aterm* – весьма обычный эмулятор терминала. Уж извиняйте!

ВЕРДИКТ LINUX FORMAT

Не более чем базовый терминал, разве что с чрезмерными визуальными эффектами.

РЕЙТИНГ 5/10

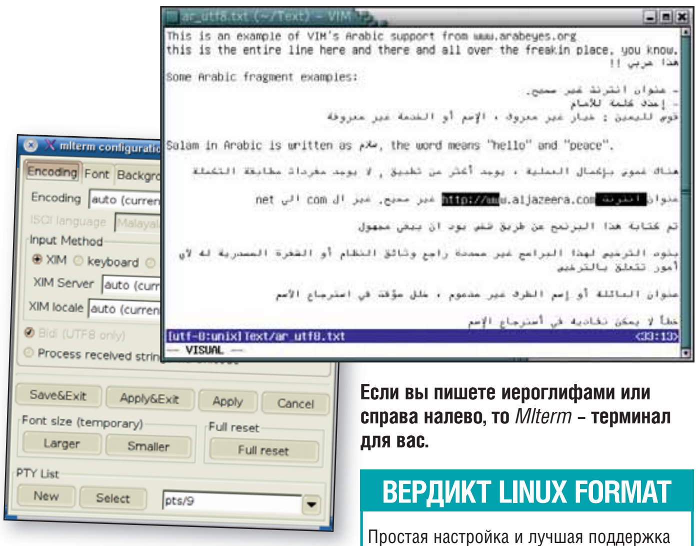
Mlterm

Исчерпывающая поддержка всех кодовых таблиц

• **ВЕРСИЯ:** 2.9.3 • **WEB:** http://mlterm.sourceforge.net
• **ЦЕНА:** Бесплатно по лицензии BSD

На первый взгляд, *Mlterm* – вылитый *Rxvt* или *Xterm*, но нажмите и удерживайте Ctrl и щелкните правой кнопкой мыши в окне – и р-раз! – он откроет конфигурационное окно *GTK2*. Число опций впечатляет – здесь имеется обычный набор шрифтов, цветов и прозрачности, но ключ к главной функции *Mlterm* лежит во вкладке Кодировка (Encoding). Этот терминал – для тех, кто использует не латиницу, и само название *Mlterm* – сокращение от «эмулятор многоязыкового терминала» (multi-lingual terminal emulator). Он имеет полную поддержку Unicode, но это еще не все. Поддерживаются сложные символы двойной ширины, подобные используемым в восточно-азиатских языках, а еще *Mlterm* работает с языками, где написание идет справа налево, например, с арабским и ивритом. Поддерживаются и индийские шрифты в кодировке ISCII, хотя авторы программы пока считают эту функцию экспериментальной.

Если вы не задали настройки сами, *Mlterm* автоматически выберет правильную кодировку на основе ваших локальных установок. Естественно, вы сможете ее изменить, если вам понадобится поработать с другим языком. *Mlterm* был написан с нуля. Производительность у него не выдающаяся и, очевидно, зависит от набора используемых символов, но если вам надо писать и слева направо и справа налево, то выбор у вас невелик.



Если вы пишете иероглифами или справа налево, то *Mlterm* – терминал для вас.

ВЕРДИКТ LINUX FORMAT

Простая настройка и лучшая поддержка языков превращают этот ничем иным не примечательный терминал в идеальный выбор для работающих с языками Азии и Ближнего Востока.

РЕЙТИНГ 6/10

X-ТЕРМИНАЛЫ
ВЕРДИКТ

Если вы сразу открыли эту страницу, чтобы узнать, какой терминал самый лучший, примите наши извинения – жизнь не всегда столь прямолинейна. Выбор терминала зависит от ваших потребностей и, в какой-то степени, от среды рабочего стола, в которой вы работаете. Если желаете, чтобы фон рабочего стола сверкал сквозь текст, можете забраковать половину рассматриваемых конкурентов. Так вышло, что ни одна из этих программ не может предложить вам прозрачности в чистом виде: только псевдопрозрачность, копирующую соответствующую часть фона рабочего стола в окне терминала (на самом деле окон под терминалом увидеть нельзя). Эта функциональность придет от *X* и самих оконных менеджеров по мере того, как вещи вроде *Compmgr* станут стабильнее.

Если вам нужен минималистский терминал без всякого украшательства, то лучшим выбором станет *Rxvt*. Если нужно нечто попривлекательнее и попроще в настройке, то лучше всего остановиться на *Eterm*. Возможно, *Eterm* понравится тем, кто работает с «минималистическими» менеджерами окон, например, с *Enlightenment*, *IceWM* или одним из вариантов *box.

Для пользователей рабочих столов-тяжеловесов, Gnome и KDE, более знакомыми, возможно, окажутся собственные терминалы рабочего стола, которые и станут для них наилучшим выбором. Да, у *Konsole* намного больше опций, чем у Gnome Terminal, но это иллюстрирует разницу между философией KDE и Gnome. Те, кому нужна возможность дополнительных настроек *Konsole*, очевидно, уже работают в KDE.

Ладно, хватит топтаться на меже. Правила требуют, чтобы был назван победитель, и им должен стать *Konsole*. Большой выбор опций и простота их использования превращают его в выдающийся терминал.



KONSOLE
9/10

Все, что нужно от терминала, и кое-что еще. Победа была бы за *Konsole* уже благодаря одному только управлению сессиями.

Запустив парочку screen-сессий, каждую в отдельной вкладке, заодно с *Midnight Commander*, причем кое-что из вышеперечисленного будет еще и работать на удаленном сервере через SSH, вы сможете оценить мощь и гибкость, предлагаемые этой программой. LXF

ДЛЯ ВАС

Мы не включили в Сравнение вашу любимую программу? Вам кажется, что сравнение терминалов – лучшее, что было со времен последнего лучшего-что-было? Напишите нам об этом на **letters@linuxformat.ru**.

ТАБЛИЦА СРАВНЕНИЯ ФУНКЦИЙ

	Aterm	Eterm	Gnome Terminal	Konsole	Mlterm	Rxvt	Wterm	Xterm	Yakuake	Yeahconsole
Скорость	4/5	4/5	2/5	2/5	2/5	5/5	3/5	1/5	2/5	1/5
Графический фон	✓	✓	✓	✓	✓		✓		✓	
Прозрачный фон	✓	✓	✓	✓	✓		✓		✓	
Темы		✓		✓					✓	
Вкладки			✓	✓		✓(1)			✓	
Профили			✓	✓						
Меню		✓	✓	✓				✓	✓	
Настройка через GUI		✓	✓	✓	✓				✓	
Выбор горячих клавиш			✓	✓					✓	✓
Текст, читаемый справа налево					✓					

(1) Воспользуйтесь *Mrxvt*, с сайта **http://materm/sourceforge.net**, для получения вкладок в *Rxvt*.







ТРИ ПАТЕФОНА

Петр Семилетов представляет вашему вниманию обзор трех аудиоплейеров.

Лет сто тому назад, желая послушать пение Федорова Ивановича Шаляпина на дому, человек шел в лавку и покупал блестящий, будто отполированный самовар, граммофон. Тогда это было дорогое удовольствие – больше полусотни рублей за штуку, да пластинки по трешке. Дешевле было выучить ноты и играть на фортепиано, или на балалайке без всяких нот. Но – песни Шаляпина или Вяльцевой, исходящие из волшебной трубы чудо-машины, оставались уделом аристократии или купцов. Покрутил купец ручку граммофона, сел за стол, узорчатой скатертью крытый, и сидит слушает, чай из блюдца попивает. Сахарок – вприкус. А Федор Иванович Шаляпин басит, басит...

Вечно только искусство, а не технические средства его передачи. Граммофоны уступили место патефонам, а те – электрофонам. На пороге двадцать первого века наступила эпоха виртуализации. Печатные машинки превратились в текстовые редакторы. Музыкальные инструменты – в VST-плагины. Магнитофоны и проигрыватели компакт-дисков тоже перешли в цифровое измерение.

В Linux первыми появились консольные плейеры: *cdp*, *mpg123*. Справедливости ради отметим, что занимают они куда меньше места, нежели граммофон, да и легче его по весу. Но – не такие красивые. Думается, поэтому возникли в Linux плейеры с графическим интерфейсом. Одним из «первых ласточек» стал XMMS, созданный по образу и подобию *Winamp*. В самом деле, зачем изобретать велосипед, если уже готовы его чертежи?

Winamp предлагал удобную архитектуру плейера. Распознавание форматов обеспечивают подключаемые модули. Вывод звука на различные устройства – опять-таки посредством расширений. Визуальные эффекты – снова они. Наконец, механизм сменных «шкурок». Это же граммофон с бесконечным набором иголок и мембран. Пользователь сам выбирает цвет «позолоты»! А ручку завода крутит таймер операционной системы.

С момента появления XMMS прошло уже почти десять лет. За это время появились десятки, если не сотни других плейеров. Прежде чем приступить к обзору наиболее интересных (на мой взгляд) из них, сделаю несколько предварительных замечаний.

В настоящее время среди плееров наметились две основные архитектуры. Первая – плейер на основе подключаемых модулей, «старая школа» а-ля XMMS. Ярким представителем таких плейеров является *Audacious*, о котором мы поговорим ниже. Второй архитектурный подход – монолитный плейер, использующий для ввода/вывода звуковых данных один из популярных мультимедийных движков. Речь идет о таких решениях, как *Xine* и *Gstreamer*. По моим наблюдениям, *Xine* более популярен в качестве движка в проектах KDE/Qt, а *Gstreamer* – в плейерах Gnome/GTK. Хотя, никто не мешает использовать *Xine* в программе на основе GTK, а *Gstreamer* – в KDE, что некоторые плейеры и делают. Функциональность *Xine* и *Gstreamer* выходит далеко за пределы воспроизведения звуковых форматов – поддерживается еще и видео, однако это уже выходит на рамки нашей статьи.

Обычно «монолитные» плейеры отличаются друг от друга только интерфейсом и набором функций утилитного характера, к воспроизведению музыки отношения не имеющего. Большинство таких плейеров оснащены мощными функциями ведения коллекции композиций, хранящейся на жестком диске, и даже святая святых – окно плейера с элементами управления – играет в них второстепенную роль. Списки песен, составленные по различным критериям, быстрый поиск нужной композиции – вот конек «монолитных» плейеров.

Audacious

Представитель старой школы

• **ВЕРСИЯ:** 1.1 • **WEB:** http://audacious-media-player.org/
• **Цена:** лицензии GPL



Плейер XMMS, как известно, написан под библиотеку GTK+1, то есть под «старую» GTK. Уже много лет как появилась GTK+2, с измененной архитектурой и более отвечающая требованиям современности. Но *XMMS* продолжал писаться под первую GTK, а со временем вообще практически перестал разрабатываться.

Время от времени предпринимались попытки портировать его на GTK+2, пока за дело не взялась команда разработчиков, поставившая себе цель создать на базе кода *XMMS* новый плейер. Плейер этот назывался *Beep Media Player*, который позже был переименован в краткое *BMP*.

Вдоволь покритиковав код *XMMS* за запутанность, команда *BMP* ударными темпами начала портировать его на GTK+2. Затем произошло вот что – разработчики *BMP* решили, что плейер надо радикально переделывать. Дескать, довольно таскать за собой звуковой движок, надо использовать *Xine* или *Gstreamer*. Разработка *BMP* прекратилась, команда взялась за *BMP*x – вначале с движком *Xine*, затем – *Gstreamer*. *BMP*x имеет очень мало общего кода с *BMP*.

Между тем, от «классического» *BMP* отпочковался проект, который получил название *Audacious*. Этот плейер в лучших традициях продолжает архитектуру *XMMS*, то есть основан на подключаемых модулях: для ввода, вывода, визуальных эффектов. *Audacious* можно расценивать как преемника *XMMS*.

Выглядит плейер традиционно – три плавающих, с возможностью «склейки», окошка – сам плейер, эквалайзер и список песен. Какие-либо функции по ведению коллекции песен и поиску в этой коллекции отсутствуют.

Число модулей в дистрибутиве с исходными текстами достаточно велико. Поддерживается MP3 (с алгоритмом декодера от mpeg123), Ogg, WMA, WAV, трекерные форматы (посредством движка Modplug), форматы музыки от игровых приставок Sega Genesis/Megadrive, NES (известна у нас как Dendy), аудио CD с цифровым считыванием, MIDI и некоторые другие форматы. Как видите, *Audacious* построен по принципу «все свое несу с собой».

Среди модулей, отличных от средств декодирования, хочется отметить Song Change, при помощи которого можно направить данные о текущей песне (например, ее название) во внешний файл. Также любопытен LADSPA host, который позволяет «навесить» на канал вывода один или несколько подключаемых модулей формата LADSPA. Их разработано огромное количество (только сотня штук лежит одним архивом на http://plugin.org.uk). Среди них всё – и эквалайзеры, и нормализаторы звука, и различные эффекты. Фактически, при использовании пакета таких расширений у вас отпадает нужда в каких-либо дополнительных средствах обработки звука в *Audacious*.

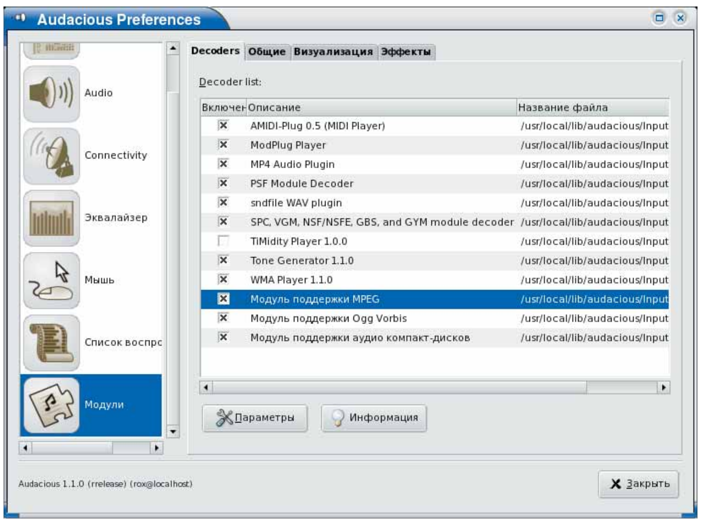
Кратко коснемся модулей вывода звука. Разумеется, в наличии и Disk Writer, позволяющий записывать воспроизводимый звук в WAV-файл. А среди современных драйверов вывода, помимо ALSA, присутствует также драйвер Jack – популярного звукового сервера, активно использующегося в программах обработки звука и создания музыки.

В плане «шкурок» *Audacious* совместим с *Winamp 2*, *XMMS* и *BMP*. Интерфейс большей частью русифицирован. А вот с поддержкой русских тэгов дело обстоит хуже. В версии 1.0 была нормально реализована возможность выбрать кодировку тэгов для MP3-файлов, и настраивалась она в опциях декодера MP3. А в новой версии плейера, функцию выбора кодировки перенесли на страницу «Список воспроизведения» окна настроек, и должен отметить, что задуманное разработчиками не работает (во всяком случае, на моем компьютере). Прописываю нужную мне кодировку – все равно русские названия отображаются «крякозяблами».

Модуль поддержки простых музыкальных CD в *Audacious* способен искать названия песен в базе данных CDDB (по умолчанию это www.freedb.org). Скачав названия дорожек диска один раз, в следующий раз плейер прочитает их уже с локального диска, то есть данные с CDDB кэшируются (в отличие от аналогичной функции доступа к CDDB в Amarok).

Однако, на момент написания этих строк, проект www.freedb.org остановлен из-за неких разногласий между его разработчиками. Хотя база данных по-прежнему доступна, неизвестно, сколь долго это будет продолжаться. В случае чего можете попробовать бета-версию нового сервиса на основе старой *FreeDB* – *freedb2.org* (http://freedb2.org/~cddb/cddb.cgi).

Итак, *Audacious* – плейер для любителей классического дизайна, подобного старому Winamp 2.x/*XMMS*, однако со множеством новых функций. С другой стороны, плейер может разочаровать тех пользователей, которые привыкли к тому, что их музыка всегда под рукой в виде автоматически составленной коллекции. Именно для таких пользователей я могу посоветовать два следующих в нашем обзоре плейера.



Gmusicbrowser

Написан на Perl и гибок в настройке

• **ВЕРСИЯ:** 0.953 • **WEB:** http://squentin.free.fr/gmusicbrowser/gmusicbrowser.html • **Цена:** лицензия GPL



Сразу признаюсь, что это мой основной плейер. После долгих душевных терзаний я перешел на него с *Amarok*. *Gmusicbrowser* написан на языке Perl и в качестве звукового движка использует *Gstreamer*, а если по каким-либо причинам это невозможно, то *Gmusicbrowser* будет воспроизводить музыку через консольные плейеры *mpg321*, *ogg123* и *flac123* (разумеется, консолей этих плейеров вы не увидите – все происходит скрытно, под сенью интерфейса на GTK+2). Обратите внимание на название первого плейера – именно *mpg321*, а не *mpg123*. *mpg321* декодирует MP3, используя целочисленные алгоритмы библиотеки MAD, которая обеспечивает отменное качество звука.

Если сравнивать его с звучанием MP3 у движка *Xine*, то в MAD звук получается более сочным, более живым. А в сравнении с *mpg123* у MAD звук более, я бы сказал, теплый. Кстати, *Gstreamer* тоже декодирует MP3 через MAD-плагин.

У *Gmusicbrowser* на первый взгляд довольно неказистый интерфейс. Исключительно стандартные элементы управления GTK+2, никакой поддержки «шкурок». Есть несколько режимов внешнего вида плейера – настраиваются они в окне настроек, на вкладке Misc – там есть опция Player windows layout со списком доступных значений. Лично мне более удобным кажется режим «with browser» (смотрите иллюстрацию). Кстати, интерфейс плейера ни капельки не русифицирован.

Gmusicbrowser умеет воспроизводить только файлы, добавленные в его виртуальную библиотеку. Составляет он ее автоматически – вам надо лишь указать, в каких каталогах искать музыкальные файлы. При этом скорость сканирования выше, нежели у *Amarok*. Доступ к базе данных осуществляется через достаточно сложную систему списков. Ведутся динамические списки по таким категориям, как исполнитель, название альбома, жанр, дата, каталог расположения и флаги.



Что за флаги такие? В *Gmusicbrowser* вы можете помечать файлы разными флажками. Есть предустановленные флаги – например, бутлег (редкая, чаще всего концертная запись, не попавшая в официальные альбомы). Можно создавать и свои собственные флаги – скажем, «Любимые песни». Кроме того, каждой песне можно выставить рейтинг.

Над панелью списка песен расположено поле быстрого поиска по нескольким десяткам критериев, причем в поиске можно использовать регулярные выражения. В настоящий момент аналогичное средство поиска планируется и для панели со списками категорий.

В *Gmusicbrowser* очень много разных кнопок, контекстных меню и информационных меток, которые разбросаны без какой-либо системы, а зачастую дублируют друг друга. Поэтому на изучение интерфейса надо потратить какое-то время, а говорить об интерфейсе трудно, потому что он у *Gmusicbrowser* многолик. Статично только окно настроек, но тут надо отметить, что многие настройки вынесены прямо в «главное окно», если в *Gmusicbrowser* применим такой термин вообще.

После некоторых изысканий обнаруживается, что плейер способен на многое. Например, он может отображать обложки к альбомам. Можете выбирать их сами из числа локальных файлов, а можете скачать из Интернета. Если *Amarok* вытягивает изображения обложек с Amazon, то *Gmusicbrowser* использует для тех же целей Google.

Кроме того, есть функция, отсутствующая в *Amarok* – можно задать картинку не только для альбома, но и для группы/исполнителя. Далее, из пункта меню Main > Open Context window можно вызывать контекстное меню, где, при условии, если у вас включен подключаемый модуль Lyrics (входит в комплект плейера), будет отображаться скачиваемый из Сети текст текущей песни. К сожалению, текст этот не кэшируется, то есть не сохраняется для повторного отображения.

В списке альбомов могут отображаться обложки. А таких списков – несколько штук. Во-первых, один список есть на панели категорий, а второй доступен под кнопкой Choose album from this artist (которая доступна не в каждом режиме интерфейса).

Список песен оснащен удобным контекстным меню, из которого можно удалять, копировать, перемещать и переименовывать файлы, а также экспортировать их во внешний файл формата **m3u**. Из того же меню можно вызвать мощный редактор тэгов. Кстати, с его помощью можно редактировать тэги не только одного, но и множества файлов одновременно. Кроме того, если выделено два или более файлов, то в контекстном меню списка песен появится пункт Mass rename, предназначенный для массового переименования файлов по заданному образцу. И еще одно замечание о редакторе тэгов: разумеется, русские тэги поддерживаются.

Выводы по *Gmusicbrowser*: из-за довольно хаотичного интерфейса этот плейер может отпугнуть часть пользователей. Однако *Gmusicbrowser* понравится тем людям, которые любят настраивать каждую мелочь в программе – он предоставляет такую возможность в полной мере. Кроме того, если разобраться в этом плейере, то вы обнаружите, что рабочие его качества – на высоте, а в количестве функций (причем полезных) *Gmusicbrowser* дает фору многим другим плейерам.

Amarok

Мы с ним уже где-то встречались...

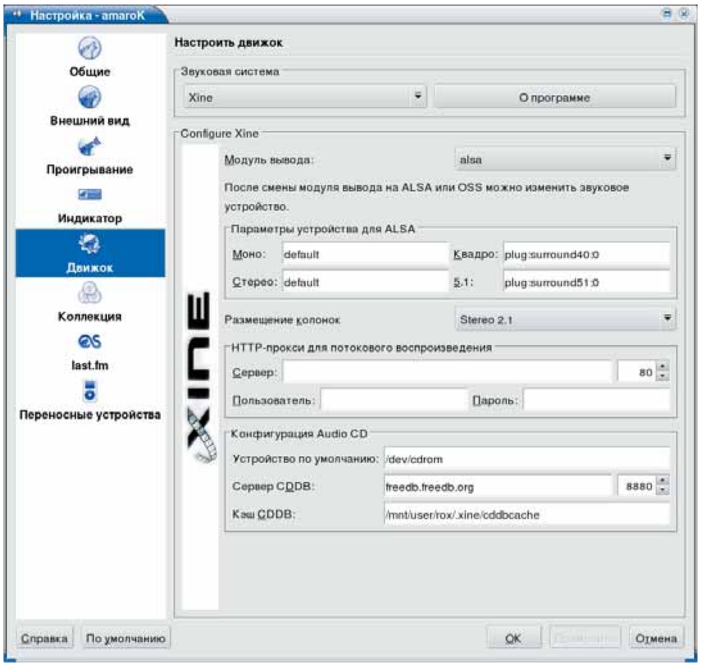
• **ВЕРСИЯ:** 1.4.1 • **WEB:** http://amarok.kde.org • **Цена:** лицензия GPL



Этот самый многофункциональный и популярный плейер для KDE еще не входит в саму среду KDE по той причине, что цикл разработки/выпусков у *Amarok* отличается от аналогичного цикла KDE. Вот такой технический момент. Разработчики обновляют *Amarok* весьма часто, и не всегда новые версии радуют пользователей – потому что добавляя новшества, разработчики почти всегда убирают что-то полезное из старого. Кроме того, с каждой новой версией *Amarok* все более зависим от внешних средств.

Например, в недавней версии 1.4.1 появилась зависимость от интерпретатора Ruby. Мало было поддержки скриптов на *Python*? Что такого не умеет *Python*, из-за чего надо было привязывать плейер еще и к Ruby?

Второй отрицательный факт. Кодировка тэгов. Жил-был не тужил *Amarok*, можно было в нем выбирать кодировку. Был такой список кодировок. И вот на каком-то этапе этот список исчез. Хотите русские тэги в *Amarok*? Только UTF-8, причем в тэгах ID3V2. У вас есть коллекция музыки, где русские тэги в кодировке Windows 1251? Конвертируйте кодировку тэгов, с помощью чего-то вроде EasyTag. Если, конечно, вы хотите видеть эти тэги нормально в *Amarok*.



Хотя я читал на форумах, что ситуация «лечится» также патчем для Taglib, однако я не пробовал. Я ведь на *Gmusicbrowser* перешел.

Но вернемся к *Amarok*. *Amarok* заведует коллекцией музыки на вашем жестком диске. Кроме того, из *Amarok* можно открывать файлы с помощью встроенного менеджера файлов, а также воспроизводить музыкальные CD (чего не умеет делать *GMusicbrowser*). *Amarok* тесно интегрирован с «прожигателем» CD/DVD – популярной программой K3b.

Плейер может представлять коллекцию в нескольких видах сортировки, а также предоставляет довольно простой поисковик (с несколькими критериями поиска). По сравнению с *GMusicbrowser* в этом плане у *Amarok* функций довольно мало. Есть, как и в *Gmusicbrowser*, динамические/умные списки песен – например, список самых новых песен или наиболее часто воспроизводимых. Можно создавать и свои «фильтрующие» списки песен.

В начале я упоминал о скриптах. Да, функциональность *Amarok* расширяется с помощью скриптов, причем плейер предоставляет встроенный графический интерфейс для скачивания новых сценариев из Сети.

Что еще умеет *Amarok*? Скачивать и отображать тексты песен, обложки от альбомов, информацию об исполнителе (из Wikipedia). Умеет взаимодействовать с подключенными устройствами – например с iPod. Встроенный в *Amarok* десятиполосный, с предусилителем, эквалайзер поможет улучшить качество воспроизведения на старых звуковых картах. В *Amarok* можно задавать «глобальные» горячие клавиши – то есть такие, которые действуют и когда окно плейера не активно.

В качестве звукового движка *Amarok* может использовать несколько библиотек – *Xine*, *GStreamer 0.10*, Helix. Если плейер собран с несколькими движками, то они становятся доступными в списке установленных движков в настройках программы.

Amarok – отличный плейер для тех пользователей, у которых мощный современный компьютер и нет необходимости видеть русские тэги во вменяемом виде. Я еще не видел лучшего плейера для KDE. И в отличие от того же *GMusicbrowser*, в *Amarok* можно работать с файлами, которые еще не добавлены в коллекцию.

АУДИОПЛЕЙЕРЫ ВЕРДИКТ



Подведем итоги. В этой статье я рассмотрел наиболее, на мой взгляд, интересные и функциональные плейеры. Они не требуют каких-либо экзотических библиотек. Все три плейера проверены беспроблемной сборкой из исходных текстов в системе Mandriva Linux 2006 – насколько я помню, для сборки плейеров я обновлял лишь TagLib (того требовал *Amarok*), хотя не уверен.

Для меня важный критерий качества кода – это легкость его сборки. В этом плане описанные мною плейеры очень хороши даже на такой относительно старой системе, как моя. Ведь, как известно, требования многих других плейеров к версиям библиотек часто идут впереди тех версий, которые в наличии в текущих выпусках больших дистрибутивов Linux, а обновление по принципу «не навреди» не всегда удается. *Audacious*, *Amarok* и *GMusicbrowser* – стабильные, обстоятельные плейеры, каждый из которых отлично справится с ответственной задачей быть «плейером по умолчанию». LXF

ТАБЛИЦА СВОЙСТВ

	Audacious	Gmusicbrowser	Amarok
Эквалайзер	да	нет	да
Визуализация	да	нет	да
Ведение коллекции	нет	да	да
Движок	Встроенный	Внешний	Внешний
«Шкурки»	да	нет	В режиме браузера
Скрипты	нет	нет	да
Плагины	да	да	нет
Обложки	нет	да	да
Тексты песен	нет	да	да

Hot Picks

НА ДИСКЕ

Лучшее на планете новинки открытого ПО!



Майк Сондерс его новехонький ноутбук с готовым Ubuntu только что прибыл. Как на Рождество!

Здесь мы проводим обзор некоторых из наиболее популярных программ в мире.

Каждый месяц мы прочесываем тысячи проектов с открытым кодом и отбираем для вас самые новые, самые изобретательные и просто отличные. Большая часть программ, попавших в обзор, записана на прилагаемый к журналу диск, но мы также предоставляем и web-ссылки, чтобы вы могли скачать новейшие версии.

Если у вас есть идеи о том, какие проекты с открытым кодом нам стоит включить в обзор, пишите нам на адрес: linuxformat@futurenet.co.uk

HOTPICKS В ЭТОМ РАЗДЕЛЕ

ОБРАТИТЕ ВНИМАНИЕ НА ПОБЕДИТЕЛЯ HOTPICKS

Все, что попадает в раздел HotPicks, заслуживает самого пристального внимания. Однако каждый месяц мы выбираем всего один самый яркий проект. Побеждает лучший!

HOTTEST PICK!

УТИЛИТА СОЗДАНИЯ CD/DVD

Bonfire

• **ВЕРСИЯ** 0.3.0 • **WEB** http://perso.wanadoo.fr/bonfire



***Gnome* для *Bonfire* дом родной, но встроенный файловый навигатор уживается и с другими оконными менеджерами.**

Непросто сделать правильно работающее ПО для записи оптических дисков. Нужно позаботиться о пользователях, делающих резервные копии, записывающих аудиодиски и создающих загрузочные DVD, либо генерирующих диски, читаемые на широком спектре устройств. С одного взгляда на страницы руководств *Mkisofs* и *CDRecord* становится понятно, насколько это сложная задача, а также почему пользователи предпочитают данные утилиты командной строки приложениям с графическим интерфейсом типа *K3b* и *GnomeBaker*. *Bonfire* – еще один игрок на поле инструментов создания дисков; его интерфейс легко осваивается новичками и обладает универсальностью командных утилит.

Поскольку *Bonfire* – приложение для Gnome, для сборки его из исходных текстов потребуется набор зависимостей. Пользователей Gnome, у которых уже установлены пакеты для разработчиков, проблемы не ожидают. Если же у вас другой рабочий стол или оконный менеджер, убедитесь, что имеются **gnome-cfs**, **nautilus-cd-burner**, **hal** и **gstreamer**. При наличии *Totem* и *Beagle* вы сможете перед записью на диск просматривать видеофайлы и выполнять сложный поиск.

Меню на выбор

При каждом старте, перед тем как запустить процесс создания диска, *Bonfire* предложит вам простое, ориентированное на выполнение конкретной задачи меню. Это гораздо более дружественный подход, чем предоставление пустого окна – вам не нужно продираться сквозь меню и справочные страницы, чтобы суметь приступить к делу. Предлагаемый выбор – создание аудиодиска, создание диска данных (например, резервной копии), копирование существующего диска или запись ISO-образа.

Когда выбор сделан, *Bonfire* снабдит вас небольшой подсказкой, описывающей, что вам делать дальше – например, накидать файлов из *Nautilus*. Превосходное решение: вся информация налицо, и вас не осаждает толпа «мастеров установки». Простейший из методов – подготовка файлов перетаскиванием их из окна *Nautilus* в список главного окна. Если вы не пользователь Gnome, можете призвать на помощь встроенный файловый навигатор.

Список файлов отображает имена вместе с размерами и типами – а также полезный полосовой индикатор внизу, показывающий, какая часть диска уже занята. *Bonfire* позволяет выбрать между 4-ГБ DVD и несколькими размерами для CD, но в нем отсутствуют некоторые редко используемые форматы, вроде двусторонних DVD и микро-CD.

Что будем делать

Итак, файлы для записи на диск подготовлены; щелчок по кнопке Burn вызовет диалог выбора устройства или имени файла (если вы просто хотите сделать ISO-образ, а записать его когда-нибудь потом). Выбор на самом деле невелик, хотя можно указать, закрыть ли диск или позволить многократную запись. Но на этой стадии хотелось бы видеть кое-что еще, в частности, опцию для загрузочного файла. Это необходимо, если вы создаете дистрибутивы или подготавливаете демо-диски Linux.

Bonfire позволяет сохранять проекты (в формате XML) для дальнейшего использования, чтобы вы могли создавать диски с небольшими вариациями без необходимости каждый раз заново воссоздавать всю структуру. Это не самое гибкое приложение создания дисков, доступное на данный момент, и ему еще нужно пройти немалый путь, прежде чем оно сможет соревноваться с такими асами, как *K3b*, но у него уже есть продуманный дизайн и простой подход, способный сделать это приложение наилучшим выбором для Gnome. В среде Gnome постоянно обсуждается вопрос, должны ли приложения Gnome включать дополнительные опции или нужно бороться за простоту. В этом случае мы бы выбрали первый вариант: функциональности *Bonfire* достаточно для выполнения базовых задач записи дисков, но он стал бы по-настоящему классным инструментом, если бы в нем были настройки для продвинутых пользователей.

КОНВЕРТОР ИЗОБРАЖЕНИЙ

SVGpage

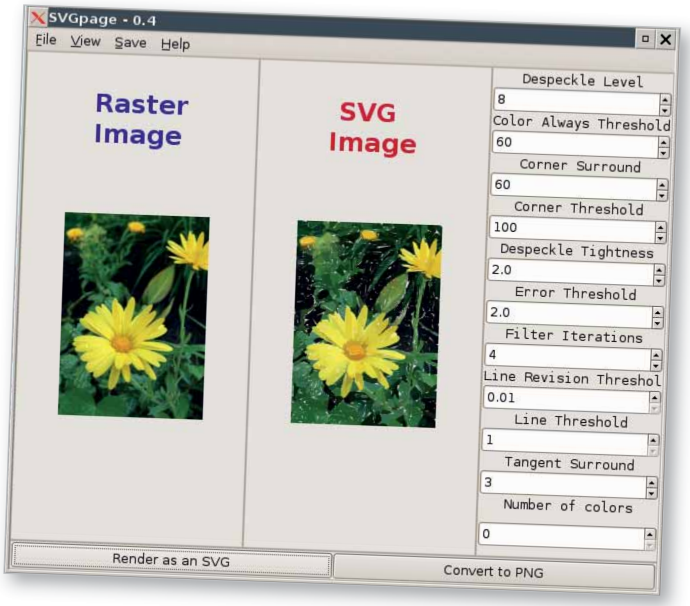
• **ВЕРСИЯ** 0.4 • **WEB** www.gimphelp.org/svgpage.htm

SVG (Scalable Vector Graphics) быстро становится стандартным форматом для векторной графики. Поддержка этого формата в Linux достаточно сильна, в частности, благодаря первоклассному редактору *Inkscape*. Многие приложения добавляют функциональность импорта/экспорта SVG. Однако преобразование растровых изображений в приемлемый векторный формат – дело не простое, если только вы не конвертируете каждый пиксел в вектор, а это уж стрельба из пушек по воробьям. *SVGpage* выполняет преобразование в SVG гораздо более гибким способом, позволяя пользователю самому выбрать уровень детализации.

Программа написана на *Python* с интерфейсом на *GTK*, поэтому вам понадобится *PyGTK*, чтобы собрать ее из исходных текстов. К счастью, разработчики предоставили статический двоичный файл, без каких-либо зависимостей. Просто распакуйте **SVGpage_bin_0.4.tar.bz2** и запустите *SVGpage* из образовавшегося каталога. Приложение откроется, показав изображение, демонстрирующее процесс конвертации из растровой в векторную графику.

Конвертация изображения очень проста: откройте файл с помощью меню File, а затем щелкните Render As An SVG внизу главного окна. Получившийся результат может оказаться далеко не впечатляющим: иногда отсутствуют детали или возникают чересчур яркие участки изображения. Но именно здесь находит применение основная функциональность *SVGpage*, индивидуальная настройка. Установить можно самые разные параметры: уровень очистки изображения, границы цветов, фильтры и многое другое. Придется поэкспериментировать; зато настроив эти параметры, вы получите для ваших изображений идеальную схему.

> «КОНВЕРТАЦИЯ ИЗОБРАЖЕНИЙ ПРОСТА.»



Эти яркие цвета и четкие формы идеальны для векторного изображения.

Особо стоит упомянуть документацию, которая хоть и не изобилует демонстрационными изображениями, но предоставляет детальные описания различных опций. Эти руководства позаимствованы у Autotrace (который обеспечивает внутренности конвертации изображений в *SVGpage*), поэтому местами они чересчур технические. Тем не менее любой человек, разбирающийся в компьютерной графике, должен без проблем с ними справиться. В целом, результаты, полученные от *SVGpage* зависят от сложности вашего изображения и выбора настроек, но при создании векторной версии логотипов и других четко прорисованных изображений программа работает просто на ура.

ТЕКСТОВЫЙ РЕДАКТОР

Medit

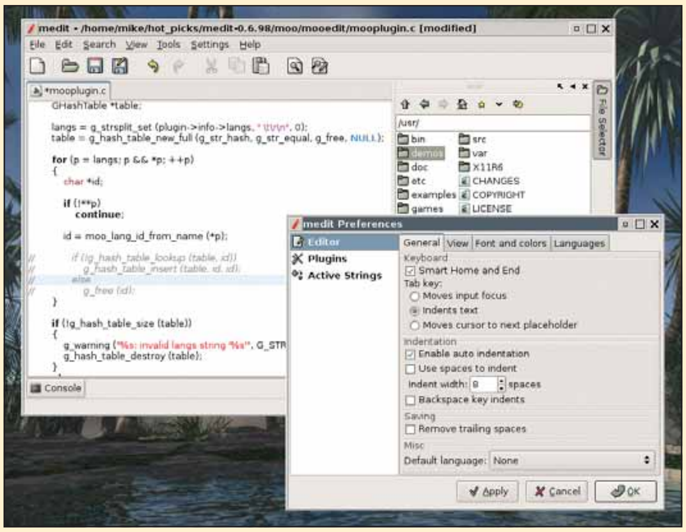
• **ВЕРСИЯ** 0.6.98 • **WEB** http://ggap.sourceforge.net/medit

Нужен ли миру лишний текстовый редактор? Для хакеров у нас есть *Vim* и *Emacs*, *Leafpad* и *KEdit* годятся для быстрых заметок, а *GEdit* и Kate занимают промежуточное положение, предлагая массу полезных возможностей. Тем не менее авторы *Medit* тоже включились в борьбу с программой, которая изначально проектировалась как компонент GGAP (**http://ggap.sourceforge.net**).

Medit задумывался как многоплатформенный (работающий под Windows и Linux) инструмент, обладающий широким спектром настроек, расширяемый с помощью дополнительных модулей, и все это с *GTK*-интерфейсом. Собрать и установить его можно обычным набором команд: **./configure**, **make** и **make install** (под суперпользователем).

Графический интерфейс *Medit* мудро использует панели с варьируемым размером и всплывающие панели, не загромождая экран и способствуя высокой продуктивности. Основное окно целиком отведено под редактор, с полезной терминальной панелью внизу экрана и со списком файлов справа. Раздражает исчезновение этих панелей, как только вы начинаете набирать текст – мы бы предпочли сообщить лично, убирать их или нет. Присутствуют все базовые операции, полагающиеся для редактора: вырезание, вставка, поиск, замена, откат и повтор.

В *Medit* немало возможностей для программистов. Главный компонент редактирования включает подсветку синтаксиса, автоматическое выравнивание, а также выравнивание и комментирование выбранного фрагмента. Разработчики утверждают, что не существует другого такого «удобного компонента редактирования текста для *GTK*» (разработчики Scintilla и обидеться могут!), и их работа заслуживает похвалы. Скорость, правда, не самая высокая, а разрекламированная функция завершения слов не работает как должно, но в остальном редактор в хорошей форме.



У *Medit* – внушительный набор опций, к удовольствию продвинутых пользователей.

Пока что это редактор как редактор, но у него имеются скрытые достоинства. Вы можете создать модули на C или Python и расширить функциональность – причем поставляется немало готовых модулей, например, вышеупомянутая панель с терминалом, и у него есть свой язык скриптов, MooScript. Добавьте сюда внушительный список клавиатурных комбинаций и широкий спектр настроек, и *Medit* решительно вступает на территорию продвинутых пользователей. Сейчас у него кое-какие мелкие проблемы со скоростью и внешним видом, но есть и все необходимое для того, чтобы стать популярным универсальным редактором для хакеров. Попробуйте его, если вы все еще в поисках идеального инструмента.

ОПЕРАЦИОННАЯ СИСТЕМА

Visopsys

• **ВЕРСИЯ** 0.62 • **WEB** www.visopsys.org

Уже полтора года мы делаем обзоры других ОС с открытым кодом. Мы рассматривали BSD-системы, Syllable и OpenSolaris, а в разделе HotPicks помещали ReactOS (клон Windows) и BeOS-подобную систему Haiku. Это самые известные альтернативные системы, но на свете полно небольших проектов, проживающих на компьютерах умников из разных стран.

Один из наиболее впечатляющих – Visopsys, графическая ОС, написанная всего одним человеком, Энди Маклафлином [Andy McLaughlin]. Хотя кодирование операционной системы в одиночку нередко встречалось в 8-битную эпоху, с тех пор компьютеры сильно усложнились. Сегодня, чтобы написать многозадачную операционную систему с поддержкой виртуальной памяти и графическим интерфейсом (или выговорить эти слова вслух), требуется много времени и таланта.

Visopsys – графическая операционная система, работающая с дискеты или с Live CD (образы и того и другого записаны на наш DVD). С момента ее первого релиза в 1997-м году она превратилась в удобную (хотя и ограниченную) ОС со своим собственным ядром, библиотекой C, командной строкой, оболочкой и графическим интерфейсом. Она поддерживает вытесняющую многозадачность и многопоточность, поддержку файловых систем FAT, ext2/3 и ISO9660 (CD) и базовые сетевые возможности. Рассматривается реализация многопользовательского доступа. Для работы Visopsys требуется 64 МБ ОЗУ и процессор типа Pentium или новее, а также мышь PS/2 (USB-устройства не поддерживаются). Видео обеспечивается обычным драйвером VESA, поэтому графика не слишком быстрая, но зато совместимая с большинством видеокарт.

Для запуска Visopsys запишите образ на CD-R и загрузите ваш ПК с этого CD. Если у вас нет CD-привода, выбирайте вариант с дискетой. После старта Visopsys спросит, хотите ли вы установить или просто запустить систему – выберите последний вариант, чтобы посмотреть, как она работает без модификаций жесткого диска.

Легкий рабочий стол

По части визуального представления Visopsys может предложить не слишком много – шрифты и иконки на рабочем столе не особо привлекательны. Возможно, вы ожидали чего-то более эстетичного от системы с таким именем, но Visual здесь имеет тот смысл, что система с самого начала задумывалась графической, в отличие от DOS, нарастившей графический интерфейс уже поверх базовой системы.

Загрузка впечатляет быстротой, а входящие в дистрибутив программы на нашем тестовом компьютере с процессором 1 ГГц запустились менее чем за секунду. Меню наверху предоставляет доступ к запущенным на данный момент программам (наподобие урезанной панели задач), а запускаются программы одним нажатием на иконку на рабочем столе. Командная оболочка, несмотря на свою рудиментарность, поддерживает завершение команд по таблице и набор Unix-подобных команд (ls, uname, ...). По сравнению с рабочими столами Linux Visopsys выглядит по-спартански: освежающая легкость и свобода от всякого мусора.

> «ОТЛИЧНАЯ ОТПРАВНАЯ ТОЧКА ДЛЯ СОЗДАНИЯ СОБСТВЕННОЙ ОС.»

А как насчет самих программ? В Visopsys входит календарь, две простые игры и программа просмотра изображений – ничего выдающегося, но достаточно, чтобы показать возможности графического инструментария. Больший интерес представляют утилиты администрирования, включающие умную утилитку разбиения жесткого диска. Эту последнюю приняли так хорошо, что Маклафлин решил выпустить специальную версию Visopsys, названную *Partition Logic*, полностью на ней фокусирующуюся. Еще здесь есть редактор конфигурации и утилита для установки системы на жесткий диск. Помимо упомянутых программ, по большому счету смотреть не на что, однако по мере расширения API мы, без сомнения, увидим больше приложений.

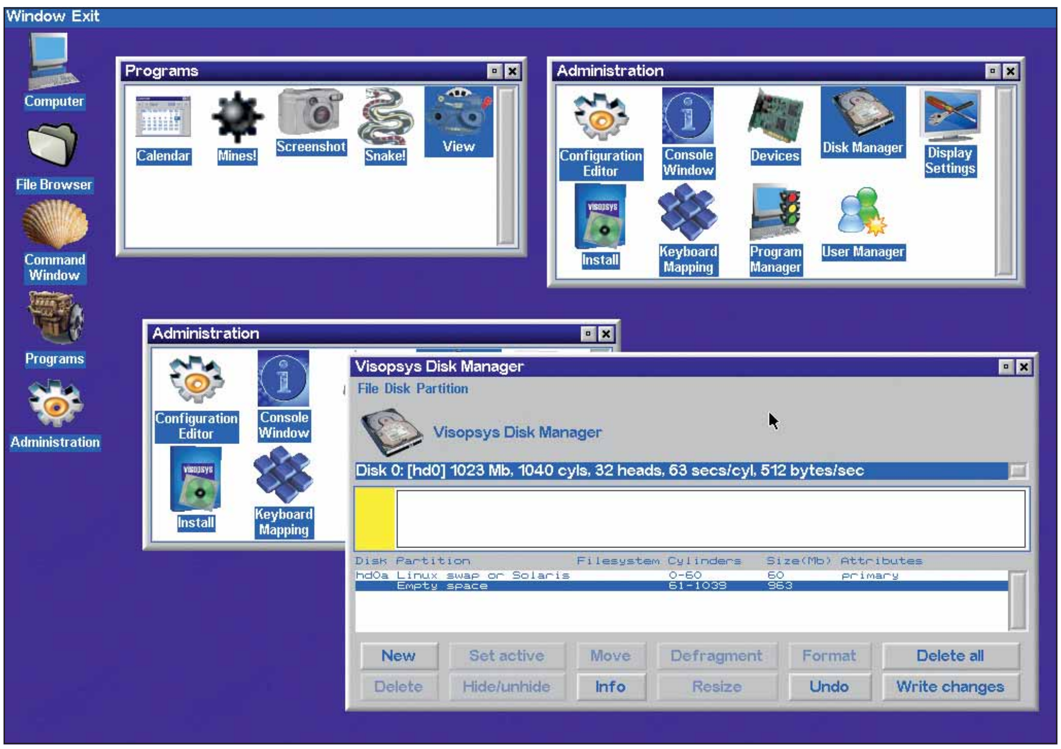
Разработчик-одиночка

Почему же мы не слышали об этой системе раньше? Пусть ей недостает возможностей распространенных систем, но это же не уменьшает интерес, например, к Syllable и Haiku. Нет, вероятно, Visopsys потому прячется в тени, что Маклафлин не ищет других разработчиков. Коль скоро это проект с открытым кодом, любой разработчик может ответвить от него свой; однако пока Маклафлин хочет развивать проект в направлении, нужном лично ему. Он с радостью выслушает ваши идеи и примет ваши исправления, но выбор направления развития системы остается за ним.

Мы считаем, что это мудрый путь, особенно учитывая ситуацию с некоторыми проектами, открытыми всем и каждому и погрязнувшими в спорах и попытках отпочковать новые ветки. Но когда время придет, хотелось бы увидеть в этом проекте больше разработчиков.

Сейчас Visopsys можно рассматривать как Minix 2006-го года: закрыта для масштабных переделок чего ни попадя, но прекрасная стартовая площадка начинающего исследователя операционных систем. Linux и клоны BSD слишком сложны для изучения системного проектирования, а компактная база исходного кода Visopsys и простой дизайн делают ее отличной отправной точкой для создания собственной ОС. В самом деле, раз эта система распространяется под GPL, вы вполне можете взять ее за основу своего проекта.

В любом случае это впечатляющий объем работы для одного человека, и проект имеет большой потенциал. За ним стоит понаблюдать.



Не ласкает взор, как OS X, зато доморощенный графический интерфейс Visopsys молниеносен.

HOT GAMES

ИГРА-ГОЛОВОЛОМКА

No Friction

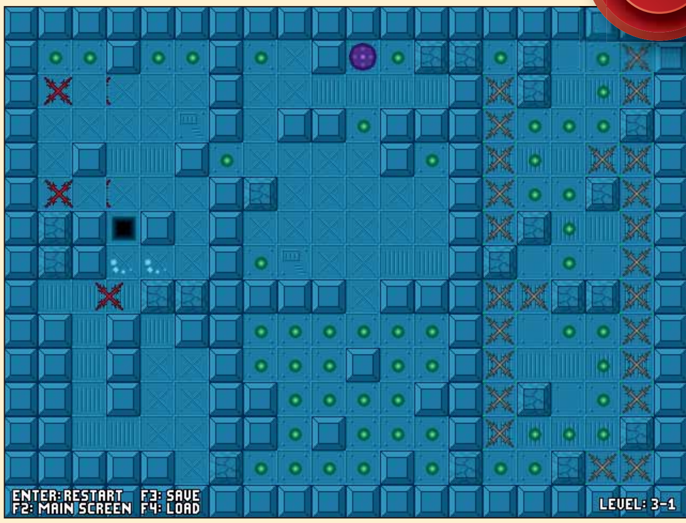
• **ВЕРСИЯ** 0.1 • **WEB** http://remar.se/fames/games.html

Здесь в здании LXF Towers мы частенько играли в *Tetris DS*. Поразительно, что эта классическая игра до сих пор доставляет удовольствие и выглядит такой же свежей, как в начале 90-х – но это и есть признак подлинной классики. Прочесывая сеть в поисках программ для рубрики HotPicks, мы были счастливы наткнуться на *No Friction* – простую на вид головоломку, которая все усложняется по мере продвижения вперед. К счастью, игре не требуется ничего, кроме вездесущего SDL, и, если пакеты разработчика SDL у вас и вправду есть, можете просто набрать **make** и затем **./friction**.

Графика не блещет, как и следовало ожидать от подобной игры – это просто дежурное блюдо: домодельные фигуры и примитивная анимация. С другой стороны, прорисовка достаточно аккуратная, а спрайты не сливаются с фоном, хотя после длительной игры постоянный синий цвет может начать давить. Игроки с художественными способностями сумеют отредактировать файлы с картинками самостоятельно – это обычные BMP-файлы, из каталога **gfx/**, и их можно открыть в *Gimp*.

Ваша задача – катить шар по арене в направлении выхода, подбирая зеленые комочки. Просто, не так ли? Но вас подстерегает ловушка: начав передвигать шар, вы не сможете его остановить, разве что он упрется в стенку. Прибавьте к этому всевозможные шипы и другие пакости, разбросанные по арене и смертельные для шара, и вам придется тщательно планировать маршрут. Кажется, до зеленого комка рукой подать, но пока вы не подойдете к нему с правильной стороны, ваш шар будет упорно натыкаться на шипы.

В игре 28 уровней, от начального уровня сложности до довольно трудных арен, требующих серьезного продумывания перемещений. К счастью, в игре имеются вариации основного игрового процесса, например, уровень, где вы контролируете сразу несколько шаров. Вы можете создать свои уровни в обычном текстовом редакторе: объекты обозначаются буквами, кроме того, существует спецификация, описывающая формат файла. В целом *No Friction* – симпатичная небольшая игра-головоломка с искусно спроектированными уровнями, заставляющая пораскинуть серыми клетками.



Пойду вниз, налево, вверх, опять налево, опять вверх… Черт, а где я был-то?

ИГРА-ГОЛОВОЛОМКА

Pipepanic



• **ВЕРСИЯ** 0.1.0 • **WEB** http://www.users.waitrose.com/~thunor/pipepanic

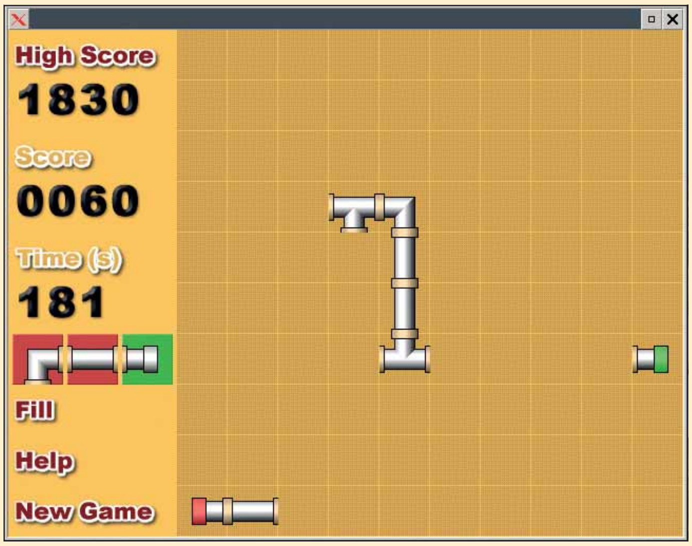
Что мы имеем из игр, посвященных прокладке труб? Вероятно, наиболее известная игра такого жанра – Mario. Затем была *Pipe Mania*, представитель классики всех времен. Возможно, эти саги о водопроводах созданы в противовес ужасам с протекающими кранами, замороженными трубами и бравыми сантехниками, заламывающими суперцены за погром у вас на кухне. Ну да в *Pipepanic* вы уж покажете, кто в доме хозяин. Эта игра создавалась под влиянием вышеупомянутой *Pipe Mania*, и, кроме SDL, ей ничего не надо, так что на большинстве дистрибутивов она соберется без проблем.

В *Pipepanic* ваша задача – соединить две трубы, используя доступные запчасти. Вы работаете на сетке размером 10x10, на которой вначале есть только зеленая стартовая труба и красная конечная.

Вам также полагается набор различных фиттингов – горизонтальные, вертикальные, Т-образные и другие – но они сваливаются к вам случайным образом, как в Tetris. Так что нельзя просто нарисовать прямую линию от начальной до конечной точки, приходится оптимально использовать наличные куски. К счастью, игра показывает вам две следующие находки, это помогает спланировать стратегию.

Тут, однако, возникает другая проблема. Соединяя начальную и конечную трубы, нельзя оставлять лишние отверстия, их нужно затыкать заглушками: если, например, в ваше сооружение добавляется крестовина, заблокируйте неиспользованные концы. После соединения начальной и конечной труб нажимается кнопка Fill, и в трубу поступает вода. Если она достигнет конца трубы – очки посыплются градом; если же обнаружится протечка, то игре конец.

Чтобы заработать по-настоящему много очков, нужно не просто искать кратчайшее расстояние из точки A в точку B, а использовать как можно больше запчастей. Именно этому свойству *Pipepanic* и аналогичные ей игры обязаны долгожительством – каждая игра непохожа на предыдущую, а по мере того как вы все чаще начинаете выигрывать не благодаря удаче, а благодаря опыту, вы начинаете понимать, что достигли определенного мастерства. Отличная штука!



Так становятся специалистами по трубам… Сначала вроде все хорошо, но скоро вас заваливает ненужными запчастями.

МЕНЕДЖЕР ЧЛЕНСТВА В КЛУБЕ

Goupil

• **ВЕРСИЯ** 0.1.0 • **WEB** http://goupil.tuxfamily.org

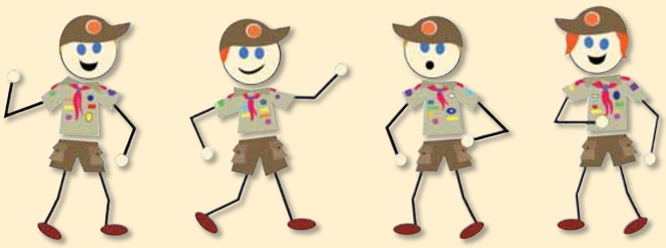
***Goupil* – приложение из категории,** до сих пор в HotPicks не затронутой: клубный менеджер. Такого рода программы не столь широко распространены, как текстовые редакторы или игры, но тем не менее они важны, особенно для тех, кто управляет клубом или сообществом. Возможно, вы решили организовать группу поддержки, или создать сайт с платным доступом, или что там еще требует учета зарегистрированных пользователей. *Goupil* спроектирован с целью максимально упростить учреждение клуба или любого группового членства и обойтись без сервера баз данных для хранения информации о членах.

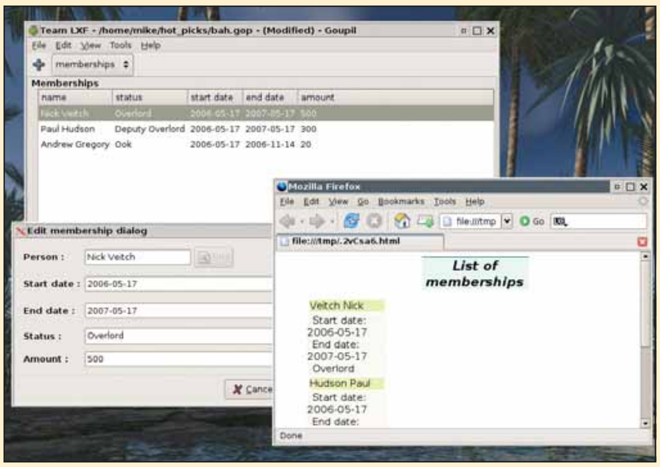
Что ему нужно, так это привязки C++ для различных библиотек Gnome. Если у вас есть Gnome и установлены его пакеты разработчика, то большая часть необходимых зависимостей у вас уже есть, но помимо этого понадобятся *libgtkmm*, *libglademm* и *libgnomevfsmm* (вы можете взять эти библиотеки из директории с зависимостями для *Goupil* на нашем DVD).

При первом запуске *Goupil* запросит у вас информацию о вашем клубе или организации – название, адрес и валюта, используемая для оплаты членства. После ввода этих данных вы попадаете в главное окно, где можно переключаться между просмотром контактов и информации о членах организации. Прежде чем добавить в организацию нового члена, нужно добавить его в список контактов; путь обходной, но работает эффективно. Во время указания членства вы можете выбрать статус (для различных уровней членства), даты начала и конца действия «членской карточки», а также сумму денег, уплаченную за вступление.

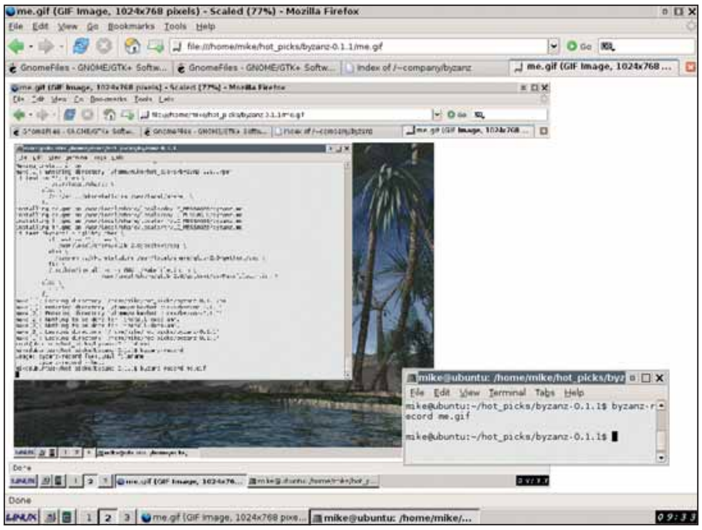
Таким образом, в основе *Goupil* – простое управление людьми, пригодное для небольшой группы членов (если организация насчитывает более 30 членов, то программа, возможно, покажется вам несколько ограниченной). Однако имеются кое-какие бонусы: например, возможность закачки контактов из *Evolution* и генерация HTML-отчетов. В качестве HTML-отчета создается базовая web-страница со списком членов и данными о каждом – функциональность не архиважная, но полезная, если вы управляете онлайн-сообществом.

Goupil пока довольно молодое приложение, и чтобы достичь версии 1.0, предстоит сделать еще многое, но основная функциональность есть, и программа уже может пригодиться небольшим организациям. Если вам нужен незатейливый способ поддержки списка членов, попробуйте ее.





HTML-экспорт из *Goupil* выглядит вполне опрятно.



Запись браузера, проигрывающего другую запись. Видите, размазано?

ЗАПИСЬ ПРОИСХОДЯЩЕГО НА ЭКРАНЕ

Byzanz

• **ВЕРСИЯ** 0.1.1 • **WEB** http://people.freedesktop.org/~company/byzanz

За несколько лет в HotPicks нам встречались программы со странными именами, и *Byzanz* – одно из самых странных. Что это имя означает? Нечто связанное с дорийским царем Византом, Константинополем или православием? На самом-то деле это программа для снятия экранных снимков рабочего стола и создания анимационных GIF-файлов, отображающих динамику его изменений в связи с вашей работой. Есть несколько причин, по которым *Byzanz* может вам пригодиться: надо сделать демо для разрабатываемой вами программы; или показать пользователям новые возможности программы; или создать пособие, обучающее пользователей правильной работе с программой (это лучше, чем пытаться воссоздать поведение программы по набору статических картинок).

Byzanz – это апплет для панели Gnome, но есть еще и утилита командной строки, на случай, если вы пользуетесь другим рабочим столом или оконным менеджером. Для сборки программы из исходных кодов понадобится пакеты разработчика Gnome 2.12 (или новее), а также *XDamage*, который в зависимости от вашего дистрибутива может быть включен в стандартный набор пакетов для X. А, например, в Ubuntu его нужно установить отдельно из пакета **libXDamage-dev**.

Пользоваться *Byzanz* так же просто, как набрать **Byzanz-record <filename>** в окне терминала; после чего начнется запись в указанный вами GIF-файл. Теперь, пока вы не нажмете Ctrl-C в окне терминала, можете использовать ваш рабочий стол как обычно, демонстрируя приложение или показывая, как нужно делать то-то и то-то. Затем GIF-файл будет сохранен на диске, и вы сможете открыть его в окне браузера. Не понравился результат? Удалите и начните запись снова.

Один из недостатков анимированных GIF-файлов, в отличие от Flash-роликов (таким путем пошли некоторые аналогичные приложения) – их размер. *Byzanz* использует расширение *XDamage*, поэтому отслеживает только те части экрана, на которых что-либо изменилось, но даже пяти секунд записи достаточно, чтобы файл разбух до 700 КБ. Лучше всего установить однотонный фон и не совершать действий, приводящих к радикальным изменениям на экране, например, свертывание и развертывание окон.

Другая потенциальная проблема – цвет у GIF-изображений 8-битовый, и, если вы демонстрируете приложение с широкой цветовой гаммой, может наблюдаться некрасивое размазывание картинки. Однако факт, что практически каждый браузер на планете дружит с анимированными GIF'ами, перевешивает эти проблемы – вашей аудитории не потребуется устанавливать дополнительные программы для просмотра.



Процесс записи запускается одним щелчком.

HOTPICKS ПОВТОРНЫЙ ВИЗИТ

РЕДАКТОР ВИДЕО

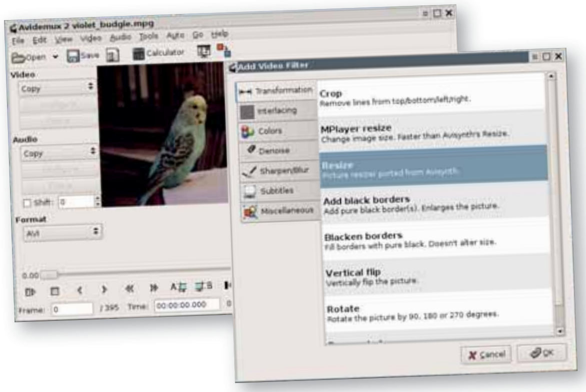
Avidemux

• **ВЕРСИЯ** 2.1.2 • **WEB** http://fixounet.free.fr/avidemux

Если вы уже давно почитываете *LXF*, то, возможно, помните наш обзор *Avidemux* из *LXF40*. Тогда это была версия 0.9rc2, теперь же перед нами версия 2.1.2, а версия 2.2 стремительно надвигается. Имя *Avidemux* – сокращение от AVI de/multiplexer: программа принимает видеопотоки, добавляет всякие фильтры, накладывает дополнительные звуковые дорожки, а затем записывает результат. Вы можете вырезать, масштабировать, смазывать изображение или делать его более резким, а также вращать видео. Программа работает с различными форматами, поддержка которых зависит от наличия установленных библиотек и кодеков (типа *MPlayer* и *DivX*).

Как вы и ожидали, со времен версии 0.9 в программу добавилось огромное количество новых функций: поддержка *Ogg Vorbis*, дискретизация звука на любой частоте, поддержка MMX/MMX2, улучшающая производительность, внутренний скриптовый движок, фильтр яркости, улучшающий видеозахват с VHS, поддержка drag & drop, поддержка импорта из PNG, мозаичные фильтры и многое другое. Большая работа проделана над интерфейсом программы, который теперь написан с использованием *GTK 2* и лучше сочетается с современным рабочим столом *Gnome* и *Xfce*.

В версии 0.9 больше всего претензий вызывал интерфейс – он казался сложным для новичков. С тех пор ситуация не сильно поменялась, однако стало очевидно, что разработчики не намерены ограничиться простыми задачами работы с домашним видео, а отдают предпочтение наращиванию функциональности.



***Avidemux* поставляется с набором фильтров для обработки видеоряда.**

Знакомым с процессом редактирования видео, микшированием и кодеками *Avidemux* предложит все, что нужно для работы, и развивается этот проект в хорошем темпе.

МЕНЕДЖЕР СПИСКА ДЕЛ

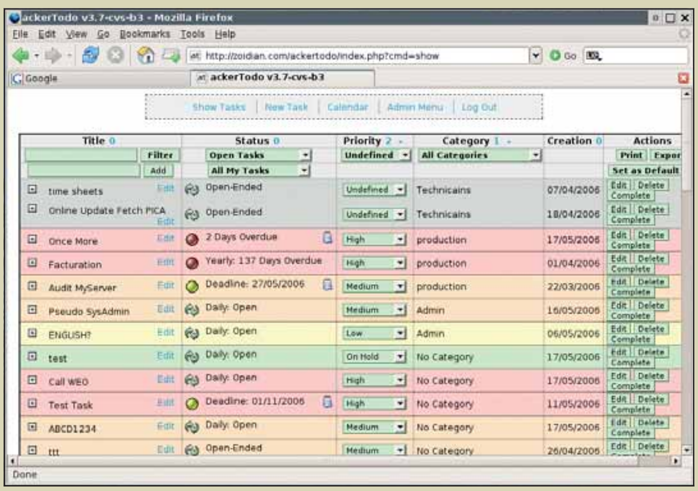
AckerTodo

• **ВЕРСИЯ** 3.6 • **WEB** http://ackertodo.sourceforge.net

Мало кто из нас дисциплинирован настолько, чтобы отслеживать более одной задачи. Спасает только ведение записей на бумаге. Поэтому совершенно логично появление множества компьютеризированных версий списков дел, и в Linux такие программы тоже есть. Одна из лучших программ такого типа – *AckerTodo*, обзор которой мы не так давно делали в HotPicks *LXF64*. Эта программа управления задачами использует web-интерфейс на базе PHP с *MySQL* для хранения данных.

Самое существенное изменение – *AckerTodo* теперь не неделимый код, а модульная программа, позволяющая самостоятельно добавлять и удалять функции; пользователям, имеющим слабое представление о программировании, это стало проще делать, поскольку не требуется погружения в глубины кода. Введена поддержка региональных форматов дат, а также оповещение через AIM об истечении срока исполнения. Вы также можете создавать задачи, повторяющиеся с заданным периодом, а из KDE позаимствована тема для иконок Crystal.

В *AckerTodo* соблюден отличный баланс между функциональностью, простотой и визуальным стилем. Приложение хорошо работало уже в серии версий 1.x, а с новыми возможностями это определенно лучший менеджер списков дел для web и для Linux. Если вам приходится жонглировать напоминалками и программами ведения записей, то это приложение может решить ваши проблемы. **LXF**



Демо *AckerTodo* можно посмотреть на сайте программы.

ТАКЖЕ ВЫПУЩЕНЫ

Новые и обновленные программы, заслуживающие внимания…

- ***SynchroEdit 0.3.9*** Многопользовательский редактор **www.synchroedit.com**
- ***SavIRC 2.0*** Кроссплатформный IRC клиент **www.savirc.com**
- ***KDiff3 0.9.90*** Утилита сравнения файлов и каталогов **http://kdiff3.sf.net**
- ***Monitorix 0.8.1*** Легковесная утилита системного мониторинга **www.monitorix.org**
- ***ALE 0.8.4*** Программа обработки изображений_ **http://auricle.dyndns.org/ALE**
- ***Outer Space 0.5.58*** Онлайновая стратегическая игра **www.ospace.net**



- ***Katoob 0.3.9*** Многоязычный текстовый редактор **http://tinyurl.com/jwjlb**
- ***Calcurse 1.4*** Текстовый персональный планировщик **http://culot.org/calcurse**
- ***Qore 0.5*** Объектно-ориентированный язык программирования **http://qore.sourceforge.net**
- ***Task Coach 0.58*** Менеджер списка дел **http://taskcoach.sourceforge.net**
- ***KPowersave 0.6.1*** Управление и мониторинг **ACPI http://sf.net/projects/powersave**
- ***Nao*** Полноценный файловый менеджер, написанный с использованием Fox Toolkit **http://nao.linux.pl**
- ***BTG 0.3*** Демон BitTorrent с текстовым и графическим GTK интерфейсом **http://btg.berlios.de**
- ***Kamefu 0.1.1*** Менеджер коллекции игр **http://kamefu.pwsp.net**
- ***Tangerine 0.2.5*** Музыкальный сервер на основе DAAP **www.snorp.net/log/tangerine**
- ***bzr 0.8.2*** Децентрализированная система контроля версий **http://bazaar-vcs.org**
- ***stalonetray 0.4*** WM-независимая системная панель задач **http://sf.net/projects/stalonetray**





МАРК ШАТТЛВОРТ В ПЕТЕРБУРГЕ

16 ИЮНЯ 2006 Г.

Литературная обработка **Алексея Федорчука** (alv@posix.ru)

ВСТУПЛЕНИЕ

Марк Шаттлворт (Mark Shuttleworth) – один из немногих разработчиков Linux, известный за пределами мира Open Source. Во-первых, его знают как удачливого интернет-предпринимателя, разбогатевшего на гребне волны «доткомов». Во-вторых, он стал вторым в истории Земли космическим туристом. И в-третьих, Марк – учредитель ряда фондов помощи слаборазвитым странам, продвижения образовательных программ в странах третьего мира и тому подобных мероприятий.

Однако в мире Open Source Марк известен, разумеется, не этим. Здесь его знают как разработчика Debian – в прошлом, и как организатора разработки семейства дистрибутивов Ubuntu – в настоящем. В этом своем качестве он возглавляет фирму Canonical – именно она осуществляет финансирование разработки всего семейства, обеспечивает распространение дистрибутива и его коммерческую поддержку, а также ведет прочую организационную работу.

Надо сказать, что дистрибутивы семейства Ubuntu (кроме собственно Ubuntu, в его состав входят также Kubuntu, Xubuntu, Nubuntu и Edubuntu) за короткое время снискали себе немалую популярность. Не в последнюю очередь – потому, что бесплатно рассылаются по всему миру, в том числе – даже в нашу страну. В результате чего в России сложилось достаточно большое и весьма активное сообщество пользователей Ubuntu.

Поэтому известие о визите Шаттлворта в Россию (Москву и Санкт-Петербург), проходившем 15–16 июня, заинтересовало в основном широкие массы узкого круга, связанного с Unix, Linux и Open Source. Тем более, что в программу посещения обоих городов входила встреча с теми, кто эти круги представляет – московскими пользователями Ubuntu и Linux вообще, и с петербургской LUG.

Московская встреча состоялась 15 июня в Институте философии РАН и организована была, насколько мне известно, компанией Altlinux. Однако на ней я не присутствовал, и потому сказать ничего определенного не могу. А вот на питерской – побывать довелось, о чем и рапортую.

Встреча с Петербургской LUG была подготовлена Линуксцентром и учебным центром «Феникс». В зале последнего, расположенном на территории географического факультета Санкт-Петербургского Университета, она и происходила. Программа мероприятия включала в себя три пункта: выступление Марка, ответы на вопросы участников и – какая встреча юниксоидов обойдется без пива! – нечто вроде фуршета.

Теоретически для участия во встрече требовалась предварительная регистрация. Однако на практике, к чести организаторов, дело оказалось гораздо проще: насколько я мог наблюдать, вход был свободный, никто ни с какими списками не сверялся, и в итоге в зале оказались все, того пожелавшие.

Надо отметить хорошее техническое обеспечение встречи. Аппаратура «Феникса» позволила выполнить аудио- и звукозапись всей встречи – не только выступления, но и вопросов, в том числе и с места. Именно обработка аудиозаписи (смею надеяться, литературная) легла в основу данного материала – в связи с чем выражаю свою признательность организаторам встречи. А фрагменты видеозаписи планируется в ближайшее время выложить на сайте Линуксцентра (http://www.linuxcenter.ru).

ВЫСТУПЛЕНИЕ

Выступал Марк на английском языке – он начал с заявления, что по-русски говорит плохо (хотя, по агентурным данным, делает это совершенно без акцента – да и сказанная по-русски вводная фраза это подтверждала). Перевод выступления обеспечивал Дмитрий Дмитриев из компании Linux Ink, известный своими работами по русификации Red Hat/Fedora и разработкой русской версии Scientific Linux. Так что суть речи Марка была доступна даже тем, кто, подобно вашему покорному слуге, английский на слух воспринимает с трудом.

Для начала Марк рассказал историю своего приобщения к Linux, ставшую в анналах Open Source уже почти столь же хрестоматийной, как история про принтер Ричарда Столлмэна или про терминальную программу Линуса Торвальдса. Один приятель дал Марку пачку дискет с дистрибутивом Slackware и шесть упаковок пива, сказав, что это – все, что нужно для освоения Linux. Правда, существует версия, что упаковка была одна – с шестью бутылками. Однако я более склонен доверять переводу Дмитрия. Действительно, без поллитры, говоря по-нашему, с Linux'ом тогда, лет десять назад, разобраться было проблематично. Так что вряд ли Марк в этом процессе обошелся даже шестью исходными упаковками...

Далее последовал рассказ о том, как зародилась идея дистрибутива Ubuntu, и об особенностях процесса его разработки. Здесь Марк подчеркнул, что одной из целей дистрибутива было достижение гармонии между стабильностью и актуальностью включенного в состав ПО. Первое достигается долговременной (трех- или пятигодичной) поддержкой стабильных релизов, выходящих через определенные промежутки времени (примерно полугодичные, хотя подготовка текущего релиза несколько затянулась). Второе же осуществимо за счет регулярных промежуточных обновлений, предназначенных для пользователей, желающих работать с самыми современными программами.

Затем в выступлении прозвучала очень интересная мысль. «Мы, дистростроители, – сказал Марк, – часто забываем, что наша роль меньше, чем роль ребят, которые собственно и разрабатывают те пакеты, что включаются в дистрибутивы. И мы должны уважать их работу – в том числе и сообщениями об ошибках, извещением о новых возможностях, включаемых в свои продукты сборщиками дистрибутивов, и тому подобными способами.»

Логическим продолжением этой мысли было высказывание об аналогичных горизонтальных связях с другими дистростроителями. В первую очередь речь зашла, конечно, о взаимоотношениях с разработчиками Debian – материнской по отношению к Ubuntu системы. Но не отвергается и сотрудничество с иными производителями – такими, как команда разработчиков Fedora – с целью обмена модификациями ядра и пакетов.

Однако и тут Марк подчеркнул, что связи, так сказать, вертикальные – с разработчиками крупных программных пакетов, таких, как Gnome, KDE, Apache, MySQL, PostgreSQL, и многих, многих других – являются более важными. Потому что в конечном счете именно их работа обеспечивает успех или неуспех любого дистрибутива.

В этом контексте прозвучал и ответ на вопрос, который меня интересовал с первого дня знакомства с Ubuntu: почему для титульного дистрибутива семейства, ориентированного, в том числе, и на начинающего пользователя, в качестве пользовательского окружения был выбран Gnome, хотя, казалось бы, KDE справляется с этой ролью как минимум не хуже. Марк объяснил сделанный выбор тем, что в момент создания Ubuntu Gnome был более простой в использовании средой, нежели KDE. Когда же разработчики KDE, оценив концепцию дистрибутива, предложили вариант со своим рабочим столом – родился Kubuntu.

Зашла речь, конечно, и о бизнес-модели, призванной сделать разработку дистрибутива коммерчески выгодной. Здесь интересен следующий момент: вместо создания единой централизованной компании фирма Canonical, обеспечивающая финансирование разработки Ubuntu и его поддержку, привлекает к сотрудничеству распределенные фирмы из разных стран – в настоящее время их более 300. Они и осуществляют регионально-ориентированную поддержку.

Маленькое отступление: как известно, Ubuntu оказался очень продуктивным клоно-породителем. Помимо всего прочего, от него происходит несколько испанских вариантов дистрибутива, ориентированных на использование в провинциальных администрациях этой страны; создается впечатление, что скоро в Испании каждая провинция будет иметь свой вариант Ubuntu. Тонкий намек: не пойти ли и нашей стране по этому пути? В этом случае востребованной окажутся и услуги фирм, способных оказать квалифицированную поддержку...

Наконец, речь дошла и до схемы разработки открытого ПО вообще и дистрибутива Ubuntu в особенности: о механизмах контроля версий и веток исходного кода, о методах совместной работы над документацией и ее переводами на разные языки – например, на санскрит (да, товарищи, в Ubuntu предусмотрена и такая локаль). Что, как было убедительно продемонстрировано, действительно, оказывается нынче ключевым моментом для любого проекта Open Source – как с технологической стороны, так и со стороны, если так можно выразиться, социальной. Впрочем, для открытого ПО эти аспекты оказываются связанными практически неразрывно – и это тоже прозвучало в выступлении Марка.

Действительно, ведь сам принцип разработки Open Source базируется на вовлечении в процесс максимально широкого круга лиц, способных к ней в принципе – и это одна сторона вопроса. Другая же, оборотная, выливается в проблему эффективности контроля над изменениями, которая может обеспечить целостность системы разработки и защиту ее от повреждения некорректно написанными фрагментами кода. То есть, попросту говоря, все сводится к тому, чтобы система была «дуракоустойчивой» – и чтобы при этом никто из разработчиков не ушел обиженным...

Вопросы разработки тесно связаны с вопросами обучения. И в планы Canonical входит создание центров обучения, тестирования и сертификации специалистов, в том числе и в России.

Последняя часть доклада, как и положено, была посвящена планам на будущее, то есть разработке следующего релиза, носящего имя Edgy Eft, выход которого запланирован на ноябрь.



ВОПРОСЫ И ОТВЕТЫ

По окончании выступления последовали многочисленные вопросы, на которые давались весьма обстоятельные ответы. Приведу их здесь, поскольку они по содержанию могут составить предмет весьма развернутого интервью. Так сказать, в коллективном исполнении.

В Мы видели очень интересный сервис для переводчиков, но он был весь на английском языке. Почему бы не сделать его мультиязычным? А то получается, что «сапожники без сапог».

О Мы решили, что сам интерфейс не нужно переводить, но некоторые части этой технологии имеет смысл локализовать. Например, это связано с обращениями по поддержке.

В Многие компании сейчас завоевывают мир простыми средствами разработки. На сегодняшний день под Linux нет простых средств разработки. Каково Ваше видение простых интерфейсов разработчика?

О Разработчики Linux пошли довольно трудной дорогой – заново изобрести велосипед. То есть установить новые правила разработки, правила открытого и свободного программного обеспечения. Мы, конечно, не можем рассчитывать на поддержку и помощь со стороны тех людей, которых привлекают легкие правила. Если Вы хотите добиться изменения ситуации, Вы должны убедить сообщество, что такие вещи, как хорошая документация, красивый дизайн, простота использования – это важные и нужные вещи. Я не думаю, что многое из того, что используется сейчас в мире коммерческой разработки программного обеспечения будет портировано под Linux. В первую очередь мы увидим перенос серверных приложений и приложений баз данных, таких, как Oracle, потому что на них есть реальный спрос. Настольные же приложения, например, *Photoshop*, имеют в Linux свои аналоги, например, Gimp, и необходимости в их переносе нет. В любом случае, не следует думать, что кто-то это сделает за вас.

В Я читал блог разработчика ядра Red Hat, который сказал, что у них очень много жалоб на то, что это работает в Ubuntu, но не работает в *Fedora*. Он сравнил ядра этих дистрибутивов, чтобы посмотреть на различия, и обнаружил в ядре Ubuntu много мелких дополнений, которые не были включены в главную ветку разработки ядра.

О Ядро Ubuntu публикуется на сайте kernel.org, так что для разработчиков ядра очень легко переносить из него все дополнения в главную ветку. Как правило, эти исправления переходят из версии в версию, но не всегда добираются до новых версий. Бывают, конечно, и случаи, что разработчик торопится и просто забывает переслать свои исправления в главную ветку.

В Можете ли Вы что-нибудь сказать относительно планов портирования Ubuntu на платформу SPARC?

О Сообщество разработчиков вокруг портирования Linux на платформу SPARC существовало долгое время, но оно не имело официальной поддержки. Однако многие пользователи хотели иметь Linux для Sparc. Так что мы сначала работали с этим сообществом, потом вышли на разработчиков собственно платформы. И теперь мы надеемся в скором времени включить SPARC в список поддерживаемых платформ Ubuntu.

В В настоящее время многие крупные компании, в первую очередь Microsoft, стараются завоевывать умы молодежи своими проектами по образованию, стажировке, трудоустройству. Есть ли у Canonical что-нибудь подобное или будет ли?

О Мне кажется, что такими вещами занимаются сообщества пользователей Linux вообще. Мы тоже считаем, что очень важно устанавливать контакты с пользователями. Но для меня самое главное, чтобы контакты происходили с людьми, которые что-то понимают в этом (смех в зале). Я думаю, что люди, сидящие в этом зале, имеют гораздо большее влияние на развитие IT, чем просто некая масса пользователей. Мы отдаем себе отчет, что начинаем с очень маленького сообщества, но это – очень образованные люди и очень эффективные.

В Планируете ли Вы включать в релизы Ubuntu ПО, которое сделает его Enterprise Ready?

О В текущие релизы включаются программы, которые люди используют в первую очередь – почта, Интернет, *Apache* и базы данных. В частности, программа установки Ubuntu предусматривает инсталляцию готового Интернет-сервера. Тяжелые сервера приложений будут включаться в дистрибутив только в том случае, если это будет востребовано сообществом. Сообщество пока не нуждается в серверах приложений (гомерический смех в зале).

В Не являясь разработчиком, я выбрал Ubuntu в качестве рабочего стола. Может быть, мой выбор неправилен, и надо было выбрать *Xandros* или *Linspire*? (смех в зале)

О Попробуйте их все – и решите, какой лучше подходит (гомерический смех в зале, переходящий в овацию).

Следует учитывать, что такие дистрибутивы, как *Xandros*, *Linspire* или, например, *MEPIS* включают в себя много коммерческого ПО, не распространяемого свободно. Ubuntu же



включает только свободное ПО и при этом работает из коробки на новом оборудовании. Кроме того, у Ubuntu очень большое сообщество разработчиков, благодаря чему ошибки исправляются быстро.

В **В последней версии 6.06 есть прекрасная поддержка свежего оборудования. Однако постоянно выходят новые устройства, видеокарты и так далее. В то же время для Ubuntu заявлена поддержка на три года для настольного варианта и пять лет для серверного. Будет ли при этом обновляться ядро для включения поддержки новых устройств?**

О Работа над текущим ядром будет продолжаться, и драйверы устройств из новых ядер будут по возможности портироваться в текущее ядро. Однако понятно, что в какой-то момент обратное портирование драйверов на ядро пятилетней давности окажется невозможным. Поэтому через какое-то время мы выпустим очередной релиз и у человека всегда будет выбор – обновлять существующую версию или перейти на новую.

В **Планируете ли Вы убеждать производителей устройств открывать код их драйверов. И что Вы предпочитаете – помогать компаниям писать драйвера под Linux или создавать собственные?**

О Наша позиция в том, что мы сможем лучше поддерживать продукты, если будет и поддержка со стороны производителя. Как правило, поставщики заинтересованы в открытии своих спецификаций, если они уверены в важности для них данного рынка. Например, я и мои друзья предпочитаем продукцию Intel, потому что это открытая архитектура, для нее быстро появляются драйверы.

В **Как Вы оцениваете взаимоотношения между разработчиками GPL-программ и разработчиками закрытого ПО? И есть ли какие-нибудь соображения по поводу BSD-лицензии, которая накладывает гораздо меньше ограничений в этом отношении?**

О Если это Ваш проект, Вы можете использовать модель двойного лицензирования, выпустив две версии – одну под GPL, другую под коммерческой лицензией.

GPL более свободна, так как она защищает свободу программы, предотвращая ее закрытие. Поэтому она предпочтительна для больших проектов

Если же Вы – разработчик небольшой программы и хотите наиболее широкого распространения своего кода, то BSD-лицензия может оказаться предпочтительней.

В **Я поставил Ubuntu только для того, чтобы отказаться от ворованного программного обеспечения. Как мне объяснить это моим друзьям? (смех и аплодисменты)**

О Я не думаю, что нужно убеждать пользователей Windows перейти на Linux, но я ищу те самые моменты, которые покажут им преимущества свободного ПО.

В **Планируются ли добавления в *Launchpad* специально для переводчиков?**

О Совершенствуются средства совместной работы переводчиков, и средства обсуждения новостей из переводов, в частности, нечто вроде "заметок на полях".

В **Полетели бы Вы в космос на корабле, все бортовые компьютеры которого работают под Ubuntu?**

О Нет. (смех и бурные аплодисменты). Знаете, какая система работает на Союзе? Восьмибитный компьютер 70-х годов, с программами непосредственно в машинных кодах, с прямым программированием памяти, в котором нечему ломаться (бурные аплодисменты).

ИНТЕРВЬЮ

В качестве завершающего штриха встречи планировалось, что Марк даст расширенное интервью для журнала LinuxFormat. Однако на большинство мыслимых вопросов ответы были получены или из выступления, или в ходе последующего обсуждения, и заставлять Марка повторять это в очередной раз было бы антигуманно. И потому все интервью свелось к обсуждению двух вопросов, показавшимся, во-первых, наиболее важными, а во-вторых, не прозвучавшим в основной части. В качестве интервьюера выступал ваш покорный слуга (хотя в итоге получилось совсем не интервью), роль переводчика исполнял Павел Фролов – генеральный директор Линуксцентра.

Сначала я поинтересовался мнением Марка на счет того, с какого конца следует подходить к пропаганде Open Source – снизу, со стороны пользователей-индивидуалов, в том числе домашних, или же сверху, от корпоративных потребителей IT. Иными словами, куда Linux придет раньше и с большим успехом, в дома, или в офисы? Ответ был достаточно дипломатичным – «не следует пренебрегать ни теми, ни другими»; но в свете отмеченной ранее, во время ответов на вопросы, ориентации на «квалифицированное меньшинство», у меня сложилось впечатление, что Марк отдает предпочтение решениям корпоративным.

Второй же вопрос касался финансовой стороны открытых проектов: должны ли они стремиться к коммерческой самоокупаемости и прибыльности, или, подобно



науке, образованию, искусству, в той или иной форме дотироваться обществом? На что Марк неожиданно ответил вопросом: «А Вы как думаете?»

Будучи, некоторым образом, представителем науки, я, разумеется, ответил, что финансирование разработок Open Source должно осуществляться по тем же моделям, что и финансирование фундаментальной науки – то есть дотационно. На что Марк сказал: «Предположим, Вы написали программу чисто научного назначения. И Вам присылают к ней патч, никакого отношения к науке не имеющий, но делающей эту программу пригодной для коммерческого использования. Включите Вы его в свою программу или нет?»

Вопрос почти поставил меня в тупик. Чуть подумав, я ответил – почему бы и нет? Ведь в сущности, наука для того и существует, чтобы ее результаты использовались. В том числе и в интересах бизнеса. Важно только, чтобы сама наука не становилась при этом бизнесом. На чем и был достигнут почти что консенсус.

Вот такое странное интервью у нас получилось.

В заключение отмечу, что встреча прошла, как говорится, в теплой и дружественной, я бы сказал – неформальной, обстановке. Не обошлась она без «раздачи слонов» – наклеек Ubuntu и последнего номера журнала Linux Format. А лично меня она навела на некоторые мысли, которыми я надеюсь поделиться с читателями в самое ближайшее время. LXF

Что такое... TENOR?

Почему вы можете что угодно раскопать в Интернете, но не в курсе, что находится на вашем собственном диске? Может ли помочь Tenor? И даст ли *Джоно Бэкон* нам какую-нибдуь подсказку?

>> Будто я не знаю, что это – толстый дядька с тонким голосом!

Нельзя же так легкомысленно отзываться о высоком искусстве оперы! Вдобавок вы заблуждаетесь: *Tenor* – довольно абстрактное понятие; в общем, это некая среда для создания контекстных сетей и построения приложений, работающих с ними.

>> Чего-о?

Я сказал, абстрактное понятие. Например, в Интернете совокупность web-страниц, адресов электронной почты и файлов определенным образом объединена: допустим, существует связь между *Mr Tambourine Man* и Бобом Диланом. Но имеет смысл также связать *Mr Tambourine Man* и The Byrds. *Tenor* предоставляет вашей настольной системе место для размещения информации такого рода.

>> То есть он просто увязывает всякие там понятия?

В простейшем случае – да, но *Tenor* способен обеспечивать более сложные отношения. Предположим, кто-то шлет вам письмо о сокращении среды обитания белого медведя в Канаде, с темой «Полярные медведи», и прикрепляет файл **F12345.jpg** – изображение здоровенной медведицы. Если вы не переименуете картинку сразу после получения (хоть оно и напрашивается, но про это легко забыть), то впоследствии отыскать ее будет трудно, поскольку поиск по словосочетанию «полярные медведи» вряд ли обратит внимание на файл с именем **F12345.jpg** (если, конечно, ваша система индексирования не отслеживает контекстные связи – например, в данном случае это изображение плюс электронное письмо, дата его отправки, размер файла, тема и т.д.). *Tenor* – система построения подобных сетей связанной информации и создания приложений, использующих эти возможности. Круто, правда?

>> Типа да, но позвольте мне переодеться хакером (фальшивая борода, сандалии и широкие штаны) и попросить вас изложить эту идею в терминах компьютерных наук.

С точки зрения науки, это направленный граф, используемый, говоря математическим языком, для отображения связей между объектами. Данная структура хранится в базе данных вместе с совокупностью свойств, в которой узлы (информация) соединены ребрами (связями). В структурные элементы встроено индексирование текста, и *Tenor* готов к этой работе, но большинство более интересных высокоуровневых вещей еще только планируется.

>> Неплохая идея. А как она появилась?

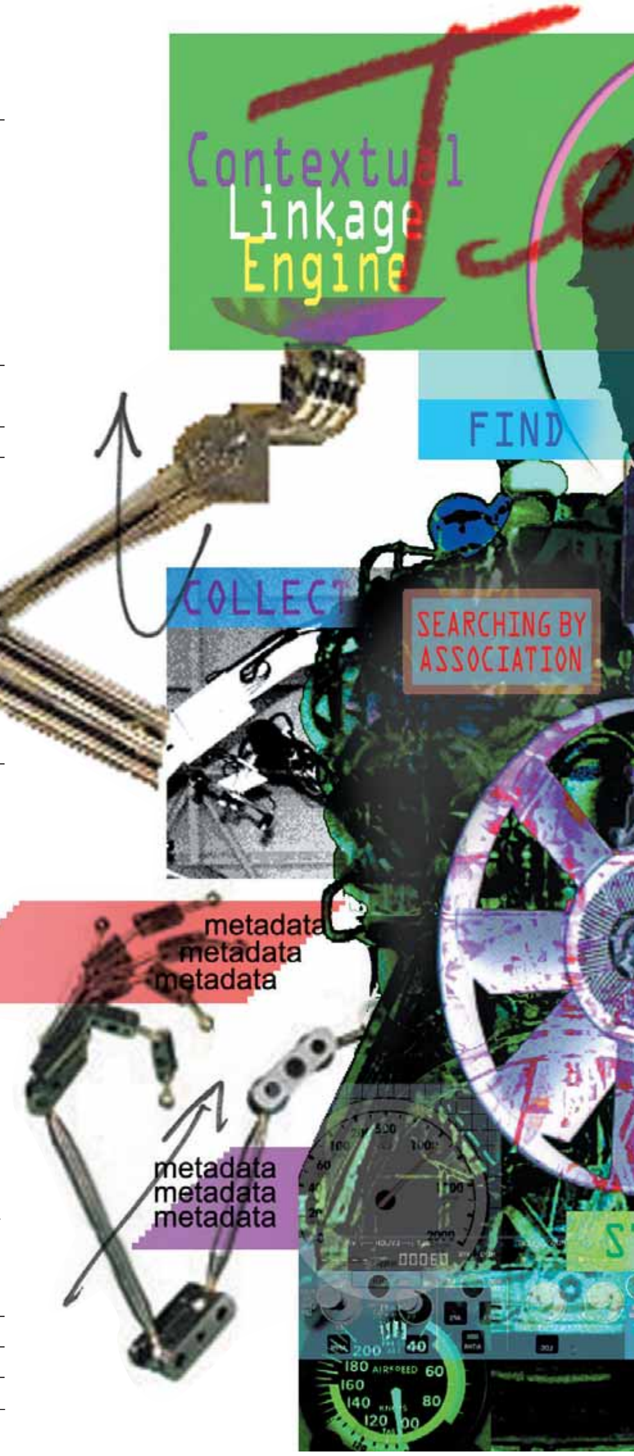
Пару лет назад Скотт Вилер [Scott Wheeler], разработчик *Tenor*'а, обнаружил, что информацию гораздо легче найти в сети, чем на рабочем столе. Он также пришел к выводу, что весьма трудно объединить беспорядочные обрывки информации в персональном компьютере. Обе эти задачи в сети решаются легко – можете привлечь свой любимый поисковый сервер и найти нечто с желаемой степенью достоверности. А на рабочих столах таких средств нет.

>> Я где-то читал, что *Tenor* имеет какое-то отношение к KDE – он только под KDE и работает?

Tenor предусматривает несколько уровней (реализован только первый). Самый низкий уровень отвечает за управление связями графов (вспомните вышеприведенное бородатое объяснение из информатики). Сюда включается создание вершин графов, их соединение, задание свойств и т.д. Он использует Qt – и весьма малую часть Qt, ее можно заменить без особых проблем. На данной основе будет сделана надстройка над KDE – средства облегчения работы с *Tenor* для разработчиков приложений. Скотт Вилер сильно привержен ко KDE и, естественно, интегрирует эти элементы в среду KDE. Но он энергично подчеркивает, что ничто не мешает кому бы то ни было написать подобные надстройки для других рабочих столов и программных оболочек.

>> Ну хорошо, а что это даст лично мне?

Преимуществ несколько. Возможно, нечто вроде *Kerry* (KDE-надстройка для *Beagle*) будет переделано для работы с *Tenor* и упростит поиск на компьютере. Но важнее то, что выражается навязшим на зубах модным словом тэгирование (навешивание ярлыков). В последнее время было много дискуссий о ярлыках и новом подходе к файловой навигации, с точки зрения управления содержимым. Примеры – динамические папки, создание ярлыков по технологии



drag-and-drop, встроенный быстрый поиск и, теоретически, файловая навигация по взаимосвязям, а не по простому местоположению в иерархии.

>> Стоп-стоп, я малость запутался...

Хороший пример – система навигации от Amazon.com, использующая немало того, что они называют увязкой контекста: информация о том, кто еще просматривал этот продукт и что он купил; рейтинги, категории; определяемые пользователем тематические списки; комментарии, которые могут служить заметками по данному товару, и т.д. Если забыть о коммерческой части и думать только о контенте, то потенциал буквально завораживает. Менеджер контента для просмотра данных по типу навигации по товарам в Amazon.com скоро станет правилом, а основу для этого закладывает *Tenor*.

>> Перейдем к деталям: что это означает для моей конкретной KDE?

Да куча идей вокруг: например, прикреплять *KNotes* к специальным ресурсам, а не разбрасывать на рабочем столе, или отследить, что музыкальный файл в *JuK* или *Amarok* прислал ваш друг посредством *Kopete*; а как насчет использования меток *DigiKam* для облегчения поиска прикреплений в *KMail*? Список возможностей огромен, но пока все это только идеи. Они должны превратиться в инструменты, чтобы стать реально полезными пользователям.



>> *Beagle* много кто хвалит, а *Tenor* на него очень похож. В чем разница?

Beagle – в большей степени классическая система индексирования текста, *Tenor* решает другие задачи. Для сравнения представьте, что *Tenor* подобен *PageRank* от Google с его системой оценки содержимого, а Beagle больше похож на AltaVista или Lycos, их подход – «давайте просто проиндексируем случайные ссылки». Без контекстных ссылок вы не сможете осуществить поиск способом, реализованным в большинстве современных алгоритмов. Другая аналогия, позволяющая понять разницу между ними – сравнение предметного указателя в книге и гиперссылок в сети. Предметный указатель в книге решает некоторые задачи, и делает это достаточно хорошо. Гиперссылки – строительные блоки для любых объектов, вы можете использовать их при создании поискового движка, напоминающего предметный указатель.

>> Как продвигается разработка?

Основные функции, для размещения объектов в графах, а также для переходов между графами и их опроса, уже в деле. Следующий шаг – создание таких функций, как поисковые классы для разработчиков, которые могут быть подогнаны под определенные приложения. Например, если вы пишете приложение для электронной почты, то, возможно, захотите включить возможности поиска, специфичные для содержимого и в чем-то отличающиеся от используемых в графическом редакторе. Существуют также приложения-помощники, без которых не обойтись. Один из примеров – диспетчер ссылок с интерфейсом D-BUS. Используя технологию D-BUS от Freedesktop.org, это приложение позволит программе, затрагивающей часть информации в графе, вызвать другую программу для употребления найденной информации. Необходимо взаимодействие различных приложений через посредство различных участков информации. Инструменты, которые облегчат внедрение *Tenor* в приложения, в настоящее время разрабатываются. Как только они будут готовы, настанет черед разработчиков использовать *Tenor* в своих приложениях. Это ожидается в серии релизов ветки KDE 4.0.

>> Я практически ничего не слышал о *Tenor*, а выглядит-то он неплохо. Разработчики живут в пещере или где?

Не в пещере, но Вилер предпочитает помалкивать о *Tenor*, пока не сможет продемонстрировать рабочую версию: ему надо заниматься разработкой *Tenor*, а не ответами на письма или дискуссиями о направлении развития неоконченной программы. Мы еле уговорили его высказаться для этой рубрики. К счастью, даже он не смог устоять перед LXF.



LXF ИНТЕРВЬЮ

«И Microsoft запросто может вступить в OSDL?» «АБСОЛЮТНО!»

Является ли Open Source Development Labs (Лаборатория Разработки Открытого Кода) центром тяжести для Linux? Выходит ли из-под контроля процесс лицензирования открытого ПО? *LXF* встретился со **Стюартом Коэном**, руководителем OSDL, для выяснения…

Задача Linux – добраться до самых разнообразных пользователей: от крупных многонациональных корпораций до владельцев домашних компьютеров, а это означает, что десятки непохожих группировок тянут его в разные стороны. Залогом невозможности монопольного захвата некой фирмой ядра Linux является работа создателя ядра, Линуса Торвальдса [Linus Torvalds], в некоммерческой организации, совместно финансируемой многими спонсорами.

Эта организация – The Open Source Development Labs (OSDL): именно в ней трудится Торвальдс и несколько других знаменитых разработчиков, и это – гарантия, что никто из поставщиков дистрибутивов не сможет принудительно направлять развитие Linux в свое русло. OSDL также выступает в роли центра, где обсуждаются стандарты, патентные реформы, лицензирование и другие темы, важные для всех проектов свободного ПО.

Пол Хадсон беседует со Стюартом Коэном [Stuart Cohen], руководителем OSDL с 2003 года, о его взглядах на сотрудничество OSDL с поставщиками дистрибутивов и с сообществом, а также на реформы в сфере предоставления патентов и лицензий…

Linux Format: Думаю, многие могли бы сказать, что OSDL тяготеет к крупным организациям. По-вашему, это справедливо?

Стюарт Коэн (СК): Многие крупные организации действительно являются членами OSDL. И немало крупных фирм-пользователей состоит в наших консультационных советах для потребителей. Мы работаем с крупными правительственными агентствами по всему миру, например, с Министерством Информационной Индустрии в Китае, ЕЭС в Европе, правительственными агентствами в США.

Да, если угодно, мы работаем с большим бизнесом, но не только с промышленными предприятиями, но и с университетами, и с государственными учреждениями. В деятельности OSDL участвует немало молодых фирм и частных лиц, не только в рабочих группах, но и в консультационных советах – мы их называем LUAC, Консультационные советы пользователей Linux (Linux User Advisory Councils), да еще некоторые частные лица вносят средства в фонд юридической защиты.

Хотя прежде всего мы концентрируемся на компьютеризации предприятий, то есть на крупном бизнесе, государственных учреждениях и университетах; я думаю, что по мере достижения успеха в этой сфере Linux будет все больше проникать в средний и малый бизнес. И это тоже будет играть для нас важную роль.

LXF: Но OSDL позиционирует себя как «центр тяжести» для Linux.

СК: Центр тяжести – любопытный термин. Когда я только начал работать, а фактически за пару недель до этого, Стив Балмер [Steve Ballmer, Microsoft] выступил и заявил, что мы никогда не добьемся успеха по причине отсутствия центра тяжести. Мы подхватили его фразу и сделали вроде расхожего лозунга: «Мы станем этим центром тяжести».

На самом деле наша миссия в том, чтобы стать местом, где могут собраться поставщики дистрибутивов, разработчики и пользователи. Это по-настоящему ускоряет темпы внедрения ПО с открытым кодом на корпоративный рынок. Определение «центр тяжести» было просто каламбуром в ответ на заявление Балмера, пытавшегося кольнуть Linux.

LXF: Я как-то говорил с одним из ведущих разработчиков ядра Linux, и тот сказал: «это не центр тяжести для Linux, это черная дыра Linux». Они считают – по крайней мере, между собой – что от крупных промышленных предприятий поступает немало средств, и эти средства исчезают где-то в недрах OSDL, а наружу выходит малая толика. Вы считаете, это правда?

СК: Я думаю, если бы вы спросили Линуса, или Эндрю Мортона [Andrew Morton], или Эндрю Триджелла [Andrew Tridgell], или других разработчиков ядра, которых мы поддерживаем или нанимаем на работу… ведь они приносят немалую пользу, и все это исходит из OSDL, с этой точки зрения.

Группа инженеров, работающих у нас над этими проблемами, довольно мала, и если представить массы людей, вовлеченных в проекты с открытым кодом или в развитие ядра, то выйдет, что только ничтожная их часть работает на нас. Но мы всегда стараемся внести достойный вклад – идет ли речь о должном лидерстве или о вкладе в кодовую базу.

У нас есть люди, занимающиеся поддержкой подсистем [ядра, – прим.ред.], у нас есть разработчики ядра, они работают у нас и играют ключевые роли, и уж для них-то, конечно, мы не черная дыра.

LXF: Итак, у вас есть Линус, Эндрю, Триджелл, а Крис Райт [Chris Wright] тоже работает в OSDL?

СК: Да, и Стив Хеммингер [Steve Hemminger].

LXF: Их число ограничено – человек десять или около того. Так оправдывает ли объем работы, выполняемый OSDL, вкладываемые средства?

СК: Ну что ж, давайте еще раз поговорим о том, что мы делаем с деловой точки зрения и о том, что мы делаем в юридическом аспекте, и что мы делаем в отношении рынка, и в образовательном и промышленном плане, и о нашей деятельности в плане техническом.



решениям, хранению данных, кластерингу и безопасности. Еще – на содержание оборудования, используемого открытым сообществом для работы над примерно 50–70 проектами, которые постоянно крутятся в нашем информационном центре.

ПРО OSDL

> «НАША МИССИЯ – СТАТЬ МЕСТОМ ВСТРЕЧИ ПОСТАВЩИКОВ, РАЗРАБОТЧИКОВ И ПОЛЬЗОВАТЕЛЕЙ.”

LXF: Куда попадает большая часть ваших средств?

СК: Наверное, больше чем куда-либо мы направляем средства в разработку – больше половины нашего бюджета.

LXF: Каков на сегодня ваш годовой бюджет?

СК: Около 10 миллионов долларов.

LXF: То есть $5 млн. в год уходит на разработку?

СК: Приблизительно. Они распределяются между разработчиками ядра и нашей испытательной группой, регрессионным тестированием и OSDL Working Set – открытым проектом, размещенным на наш web-сайте, который следит за программами и библиотеками, которые люди запускают поверх ядра. Еще эти средства идут на финансирование рабочих групп, занимающихся телекоммуникационными или настольными системами, или вычислительными центрами. А еще – тратятся на инициативы по сетевым решениям, хранению данных, кластерингу и безопасности. Еще – на содержание оборудования, используемого открытым сообществом для работы над примерно 50–70 проектами, которые постоянно крутятся в нашем информационном центре.

Так что с инженерной точки зрения мы занимаемся очень многим. А еще у нас есть небольшая ИТ-группа, которая не только осуществляет поддержку нашей внутренней деятельности и поддержку проектов, связанных с двумя нашими информационными центрами – один в Токио, один в Бивертоне, но также вносит вклад в большое количество всевозможных проектов с открытым кодом.

LXF: И все-таки около половины приходится на инженерную часть?

СК: Правильно.

LXF: Сколько инженеров вы за последнее время уволили, если, конечно, увольняли?

СК: Пару инженеров, когда мы увольняли несколько человек [летом 2005], чтобы высвободить средства для областей, на которых мы решили сконцентрировать внимание – глобальная экспансия, IP-деятельность [IP – intellectual property – интеллектуальная собственность, – прим. ред.]. >>

<< Расширение происходит в сфере маркетинга, развития бизнеса, ИТ, финансов…

LXF: Что вы называете «IP-деятельностью»? Термин широк!

СК: Например, сейчас ведется разнообразная деятельность в юридической области. Имеются проблемы с авторским правом, явная проблема с иском SCO, а несколько лет назад мы объявили о фонде юридической защиты для поддержки инженеров, к которым предъявляет претензии SCO. Менее чем за неделю мы собрали около $3 млн. для содействия тем двум конечным пользователям, и это отлично сработало, поскольку SCO больше не преследует конечных пользователей, о чем заявлено публично.

Мы много делаем в области лицензирования и в области переизбытка лицензий. Как вам известно, на сегодняшний день между FSF и OSI имеется более 60 открытых лицензий, но мы полагаем, что большую часть необходимой работы вполне могут выполнять где-то полдюжины лицензий. Так что мы очень серьезно этим занимаемся.

Мы также занимаемся вопросами торговых марок, в частности, касающимися торговой марки Linux и Linux Mark Institute, созданного Линусом много лет назад. Вместе с Ларри Огастином [Larry Augustin] я вхожу в совет директоров Linux Mark Institute, мы следим за составлением сублицензий торговой марки Linux, наша задача – убедиться, что торговая марка сохраняется и поддерживается и становится неотъемлемой частью всего происходящего процесса, словом, следить, чтобы торговая марка Linux было защищена.

LXF: Но вы ведь создали Фонд Поддержки IP (IP Support Fund)?

СК: Да, мы собрали средства для двух целей – для вопросов по патентам и ИТ, решаемым у нас внутри, и для поддержки Software Freedom Law Centre, основанного Эбеном Могленом [Eben Moglen], мы вложились в этот Центр. Вы могли видеть, что за несколько лет мы выделили Эбену $4,2 млн. на оказание юридических услуг открытым проектам, которые мы сочли ключевыми. Поскольку эти проекты включали в себя все больше и больше программ, которые использовались на крупных предприятиях, в крупных университетах, крупных правительственных учреждениях, мы хотели надежно укрепить их юридические позиции.

В свою очередь, мы тоже принимаем участие в этой деятельности; плюс к тому, Эбен собирается заниматься одним из принципиальных моментов – пересмотром GPL. И пока он осуществляет миграцию с GPL 2.0 на 3.0 через фонд Free Software Foundation, нам надо обеспечивать существование Software Freedom Law Centre и его возможность этим заниматься.

Это все только по поводу аспекта IP. Но есть и еще одна сфера серьезного внимания – вопрос «Чем тормозится распространение Linux?». Некоторые разработчики могут сказать, что все дело в коде, но есть многое и помимо кода, поскольку на данный момент с технической стороны Linux достаточно хорош для массового использования по всему миру. Проблемы в основном лежат как раз в сфере бизнеса, это проблемы, касающиеся совместной работы программ с открытым кодом в Linux, поскольку все это завязано на стандартизации дистрибутивов, уверенности в их совместимости и надежности, как в США, так и во всем мире.

ПРО МЕДЛЕННОЕ РАСПРОСТРАНЕНИЕ

«Linux достаточно хорош технически для использования по всему миру.»

LXF: А вам не кажется, что, возможно, лучше оставить вопросы стандартизации на LSB [Linux Standard Base, База Стандартов Linux – прим. пер.], вопросы интеграции – на поставщиков дистрибутивов, а OSDL будет работать в тех областях, для которых предназначена?

СК: Никоим образом. У нас есть совет директоров, у нас есть бизнес-план, у нас есть согласованная программа нашей деятельности. Поэтому мы полагаем, что LSB – это очень важный первый шаг, но чтобы эффективно работать с дистрибутивами во всем мире, ей надо стать намного сильнее. Когда вы рассматриваете деятельность, связанную с Asianux, Mandriva и Debian, вы рассматриваете эталонную архитектуру из Китая и проекты Open Source Symposium из Китая, Японии и Кореи. Мы считаем очень важной сильную совместимость между Red Hat и SUSE и основной массой дистрибутивов из разных стран. Именно это увеличивает ценность и повышает доверие потребителей. Мы делаем намного больше, нежели простое написание кода.



LXF: Значит, инженерам пришлось уйти, чтобы высвободить фонды.

СК: Повторяю еще раз, это было лишь несколько человек.

LXF: Сколько инженеров у вас работает?

СК: Всего у нас более 50 сотрудников. Несколько человек ушли, и у нас осталось 40, и, как я уже сказал, это касалось всех наших подразделений.

LXF: Поговорим о патентах. Это область, где OSDL стоит по обе стороны забора.

СК: Это как?

LXF: Ну, большая часть ваших средств поступает от фирм-владельцев многих патентов. Многие из них твердят: «Нам не нравятся патенты, надо бы их отменить, но мы будем все равно их регистрировать». Например, IBM ежегодно регистрирует 3000 патентов, или похожее безумное число. Вы не думаете, что OSDL ведет себя противоречиво?

СК: Отнюдь. Я думаю, ваши слова отражают реальность. Посмотрите на патентные бюро по всему миру, будь то Китай, Япония, ЕЭС, США – все они ждут патентной реформы.

Мы считаем патентную реформу неплохой идеей, нам кажется, что патентов на ПО слишком уж много. Многие из них недействительны, многие просто не следовало выдавать. Но реальность такова, что переход из сегодняшней ситуации в такую ситуацию, когда вообще не будет патентов на ПО, займет длительное время. И в течение этого переходного периода мы хотим быть уверенными в том, что ведется необходимая деятельность, чтобы разработчики продолжали работать, продолжали обновлять и разрабатывать код, без всяких проблем, связанных с патентами.

Пусть вы слышите о том, что Red Hat предоставляет патенты [в свободный доступ, – прим.ред.], или HP извещает о защите, или IBM “выкладывает” свои патенты – а мы считаем, вот и хорошо. Возьмем Sun: эта фирма выдала множество патентов, но только пользователям Solaris. Было бы очень хорошо, если бы существовало широкое соглашение о патентах на открытый код, если угодно.

LXF: Соглашение о предоставлении патентов Nokia касалось только ядра. А у вас ведь Лаборатория Разработки Открытого Кода, а не Лаборатория Разработки Ядра (Kernel Development Labs). Неужели вам этого хватает?

СК: Нет, но еще раз повторяю – это лишь хороший первый шаг. Нельзя же ожидать, что все фирмы утром проснутся, соберут

ПРО ЧЛЕНСТВО В OSDL

«Некоммерческой организации, нейтральной к поставщикам, нельзя быть разборчивой: кого принимать, а кого нет.»

свои патенты и отдадут их. Однако все эти фирмы пытаются работать с теми бизнес-моделями, которые сегодня актуальны. А правительства пытаются реформировать патентную политику. Но быстро такие дела не делаются.

Причем те же фирмы стараются убедить разработчиков всего мира, что их патенты не будут использованы против них ни в каком виде или форме, и что у разработчиков есть полный доступ к этим патентам по части разработки ядра или инноваций.

Ну и что, значит, завтра все патенты будут полностью отменены? Нет. И я не знаю ни одного места в мире, где это в самом деле может произойти – разве что у кого-то в воображении.

LXF: Но вы же говорили, что большинство предоставленных патентов недействительны. Вы не сказали, что недействительны все патенты. То есть, на ваш взгляд, патенты на ПО выдавать можно, но только после тщательного изучения?

СК: Вы говорите о разных вещах. Я действительно верю, что через какое-то время реформа патентов добьется полной отмена патентов на ПО.

LXF: Так вы этого хотите?

СК: Да. Мы думаем, что именно так и произойдет. Но не сейчас. Если заглядывать вперед, то мы считаем патентную реформу правильным направлением. Через много, много времени все правительства в мире коллективно придут к этой точке зрения.

Но оглянитесь назад и посмотрите на имеющиеся патенты на ПО: большинство из них было недействительно с самого начала. Называют разные цифры – треть, две трети, 80% – разные люди приводят разные данные по поводу их количества. То есть я хочу подчеркнуть: задним числом самые разные люди вам скажут, что большинство патентов недействительно, и их не следовало предоставлять вообще.

LXF: Хочу уточнить: итак, точка зрения OSDL на самом деле в том, что патенты на ПО выдавать не надо.

СК: Наша... Чтобы вам стало полностью ясно: патенты на ПО отживают свое, но они отживают свое с течением времени. Я не собираюсь здесь провозглашать, что нужно отменить все патенты уже завтра, и потом весь остальной мир сообщит, что на это потребуется 25 лет, а вы заявите: «Вот видите, OSDL ошиблась».

LXF: Ну, я не этого добивался. Я просто говорю, что вы верите, что в какой-то момент в будущем – необязательно завтра, может быть, через 25, 30 лет – патенты на ПО будут отменены, и вы этого хотите.

СК: Я думаю, что к этому движется весь мир, и мы это будем поддерживать.

LXF: А вы не рассматривали возможность каких-либо санкций против тех членов, которые игнорируют эту позицию?

СК: Мы – некоммерческая организация, нейтральная по отношению к поставщикам дистрибутивов. Так что в какой-то степени это трудно…

LXF: Но это же не значит, что вы впустите к себе кого попало?

СК: На самом деле значит. По закону.

LXF: По закону вы обязаны позволить вступить в свою организацию любому?

СК: Да. Неужели вы думаете, что можно работать в некоммерческой организации, нейтрально относящейся к поставщикам, и при этом выбирать – кому позволить вступать в нее, а кому нет?

LXF: Неужели нельзя?

СК: Вот именно, нельзя.

LXF: Ну хорошо, если бы вы работали в благотворительном фонде Спасения Детей и в него захотела бы вступить некая организация, являющаяся частью движения «Смерть детям», вы же не пустили бы их в ваш фонд. Вы могли бы им сказать: «Вы не разделяете наших взглядов».

СК: Дело в том, что есть законы, регулирующие деятельность некоммерческих организаций. И конечно же, мы не захотим, чтобы членом OSDL стал тот, кто не разделяет наших взглядов на миссию, стратегию и направление.

LXF: Вот-вот. Значит, кому-то вы можете запретить членство?

СК: Вопрос не в том, чтобы запретить, вопрос звучит иначе: «Вот наша миссия, цели и задачи – согласны ли вы с ними и готовы ли их поддержать?». И если нам ответят «да» – мы готовы принять их.

L***LXF:*** А если нет?

СК: Тогда они и вступать не будут. Но препятствовать им мы не можем. Мы не можем переписать закон.

LXF: Значит, Microsoft запросто может вступить в OSDL.

СК: Абсолютно. У нас есть еще 70 фирм – членов OSDL, и их количество, возможно, перевалит за 100.

LXF: И сколько же этих фирм, по-вашему, будут конфликтовать с вашей точкой зрения на патенты?

СК: Вероятно, подавляющее большинство из них работает над патентной реформой. Не думаю, что есть хоть одна фирма, которая бы не работала – хоть в какой-то степени – над реформой патентов на ПО. Вопрос только в темпе и степени, но в общем любая фирма, вступавшая в OSDL, заявляла, что патенты должны быть реформированы.

LXF: Я думаю, что в США патентная реформа отличается. Мне кажется, что некая разновидность патентной реформы происходит прямо сейчас: это попытка уменьшить опасность судебных исков. Получить патент все так же легко, но предъявить по нему иск уже труднее. Здорово, конечно, что IBM или кто другой под это подпадают, но от этого мало прока с точки зрения радикальной отмены патентов на ПО.

СК: Да, могу еще раз сказать: я думаю, это долгий процесс. Возьмите Бразилию, Китай, Японию, ЕЭС и США – уж не будем рассматривать каждое патентное бюро в мире. Даже основные патентные бюро занимают разные позиции по патентной реформе.

Отстает ли правительство США? Да. Движутся ли они медленнее, чем некоторые другие патентные бюро? И это правда. Но мне думается, что в целом все двигаются в нужном общем направлении. **LXF**

Читайте полную онлайн-версию интервью на **www.linuxformat.co.uk/cohen.html**, где Стюарт отвечает на вопросы о SCO, BitKeeper и сходстве с Coca-Cola.



Linux vs Vista: наши бьют!



Блистательный рабочий стол с 3D-ускорением будет править миром... о да, это про Linux. **Пол Хадсон** разбирается, почему «Linux против Vista» – это битва, которую мы можем выиграть, даже ничего не делая.

НА ДИСКЕ

- *Beagle 0.2.6*
- *Compiz*
- *SuperKaramba 0.39*
- *Xgl*



Выход Windows Vista ожидается в конце этого года... после значительной задержки.

Windows Vista вносит ясность в ваш мир, чтобы вы могли безопасно и легко выполнять повседнев-ные задачи и немедленно находить, что хотите, на своем ПК. По крайней мере, такую установку Microsoft навязывает миру, и, если вы пользователь Windows, для вас это правда – как минимум, отчасти.

Но вы читаете этот журнал как раз потому, что вы – не пользователь Windows, и не удивитесь, узнав, что многие из «инноваций», которые Microsoft заявляет для Vista, давно доступны в Linux. Фактически, Linux даже обладает такими функциями, которые не включили в Vista, побоявшись сорвать сроки выхода! Все мы знаем людей, до сих пор зацикленных на Microsoft, и у многих из них капают слюнки при мысли о новом релизе Windows.

Так что в этом выпуске мы предоставляем вам оружие для проповедей, которое поможет выигрышно показать ошеломляющую крутизну Linux и убедить ваших использующих Windows друзей перейти на него. Все, кто объявлял Linux неподходящим для настольных систем, убедятся в своей неправоте...

НАЗАД, В БУДУЩЕЕ

Бывалые линуксоиды сразу узнают некоторые «новые» функции Vista:

«Инновации» Vista	Linix-эквивалент
Instant Search	*Beagle*
Интерфейс Aero	*Xgl*
Виджеты рабочего стола	*SuperKaramba*
Network Explorer	Bonjour
Встроенный брандмауэр	Встроенный брандмауэр
BitLocker	Встроенное шифрование файловой
Браузер *IE7* со вкладками	Браузер *Firefox* со вкладками
Графическая библиотека *Avalon*	Графическая библиотека *Cairo*
Графические интерфейсы на XAML	Графические интерфейсы на XUL
Автоматические обновления	Автоматические обновления

Мгновенный поиск файлов

Пользователи, перешедшие с Windows 2000 на Windows XP, пережили одно из самых пессимистических обновлений за все время: Microsoft переименовала меню «Найти» (Find) в «Поиск» (Search). Сочувствуем: попытка найти скачанные файлы или установленные программы – непростая задача для многих операционных систем, во многом из-за чрезмерно разрастающихся жестких дисков. Но с выходом Vista Microsoft обещает прорыв: инструмент *Instant Search*, «поиск, интегрированный в рабочий стол». Знакомо, а?

В Linux мы привыкли к собственному инструменту настольного поиска – *Beagle*, который индексирует документы, электронные письма, диалоги обмена мгновенными сообщениями, изображения, звуковые и видеофайлы, приложения... и даже вашу историю посещения web-страниц. Вся эта информация немедленно отображается в одном окне поиска.

Beagle следит за всей вашей файловой системой и засекает, когда файлы изменились. Несколько лет назад эту работу выполняла библиотека *libfam* (file alteration monitor, монитор изменения файлов), она бормотала себе под нос: «Хмм... что-нибудь поменялось в **/usr**? Нет? Ладно. А сейчас? Нет? Славненько», и так далее – *она наблюдала за каждым каталогом*.

Нетрудно догадаться, что *libfam* не отличалась бешеной скоростью, но теперь у нас есть решение: *Inotify*. Это компонент ядра Linux (которое, собственно, и выполняет запись данных на жесткий диск), сообщающий программам об изменении файлов. Процессор практически не нагружается, а *Beagle* полностью сканирует систему всего один раз (во время первоначальной установки), после чего только делает обновления по мере изменения файлов.

Друг человека

Магия *Beagle* [англ. «гончий пес», – прим. пер.] заключается не просто в поиске по именам файлов в соответствии с вашим запросом: интеллектуальные фильтры обрабатывают различные типы файлов индивидуально. Например, у документов *OpenOffice.org* или PDF *Beagle* читает содержимое (т.е. текст, набранный на странице, листе электронной таблицы или слайде). У звуковых файлов читается ID3-тэг, хранящий сведения об исполнителе, название альбома и другую полезную информацию. У электронной почты индексируется тело сообщения, а также тема и информация об отправителе и получателе. Иными словами, *Beagle* извлекает из каждого файла наиболее полезные данные.

Всегда рядом

Beagle установлен и доступен по умолчанию как в Fedora Core 5, так и в SUSE 10.1, и многие другие дистрибутивы включают его как опцию. Еще более впечатляет факт, что Gnome 2.14 – окружение рабочего стола, стандарт трех самых популярных дистрибутивов – использует *Beagle* как низкоуровневый поисковый механизм (back-end) для основного файлового менеджера *Nautilus*.

Эта новая версия Gnome уже пошла как стандарт в Fedora 5 и Ubuntu 6.06, то есть вы можете нажать Ctrl+F в любом окне *Nautilus* (или на рабочем столе), и ваши файлы мгновенно найдутся. Более того, результаты поиска автоматически обновятся, если вновь созданный файл или поступившее сообщение подпадут под критерии поиска.

> «BEAGLE НАБЛЮДАЕТ ЗА ВСЕЙ ФАЙЛОВОЙ СИСТЕМОЙ, ОБНАРУЖИВАЯ ИЗМЕНЕНИЯ.»

Сочтя, что результаты поиска в дальнейшем вам пригодятся, можете сохранить их в виртуальной папке, а затем повторять поиск, просто дважды щелкнув по иконке сохраненного запроса – результат тут же вберет изменения вашей файловой системы. Это великолепно! Если вы раньше никогда не пробовали *Beagle* в работе, взгляните на врезку внизу...

>>



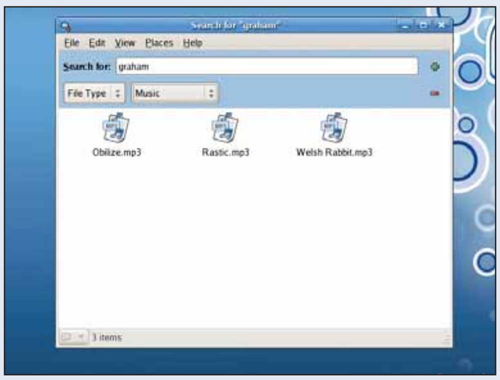
ВНАЧАЛЕ БЫЛ LINUX... **BEAGLE**



Чудеса *Beagle*: мы пошарили по файлам в поисках слова graham и нашли музыку, сочиненную Грэмом Моррисоном, плюс сайты, на которые мы заходили и где он упоминается.



Не закрывая окна поиска, мы скачали несколько рисунков Грэма из блога *LXF*. Beagle сразу же обнаружил их и добавил в наши результаты.



Beagle с легкостью обеспечивает «прозрачный» поиск из других мест вашего рабочего стола, например, из *Nautilus*.



Если какой-либо запрос выполняется довольно часто, сохраните его как виртуальную папку и просматривайте его результаты просто двойным щелчком.

Даешь шикарную графику!

Одно из самых разрекламированных нововведений Vista – новый графический интерфейс пользователя Aero, который добавляет трехмерную оболочку к традиционному внешнему виду и ощущениям XP. Довольно мило – но ему далеко до аппаратно ускоряемого интерфейса, включенного в Mac OS X несколько лет назад.

И даже взяв все лучшее от Vista и OS X, вы и близко не подойдете к Linux, благодаря самому замечательному графическому нововведению на сегодняшний день – *Xgl*.

3D: Дождались

Разработанный и переработанный за последние несколько лет, *Xgl* наконец-то выполнил обещание предоставить графику с настоящим аппаратным ускорением. Раньше прорисовка всех ваших окон была заботой процессора, и это отнимало весьма значительные ресурсы. С *Xgl* весь процесс прорисовки окон на экране выполняется вашей видеокартой, она делает это намного быстрее. Вдобавок открываются новые возможности: ваша видеокарта умеет рисовать отнюдь не одни прямоугольнички и кружочки, так почему бы этим не воспользоваться?

> «XGL РЕАЛЬНО ОБЕЩАЕТ АППАРАТНОЕ УСКОРЕНИЕ ГРАФИКИ.»



Xgl натягивает каждое окно с вашего стола на многоугольник как текстуру, и затем помещает его в отдельном слое поверх других в ОЗУ, применяя мягкие тени, сглаживание пикселей и направленное освещение. Каждый из ваших виртуальных рабочих столов затем проецируется на грани куба, который можно вращать в реальном времени. Вы больше не увидите «разрывов» графики при перемещении окон по экрану; и можно даже добавить эффект тряски.

Никто – вот именно, никто – не сможет остаться равнодушным, увидев *Xgl* в действии. Несмотря на молодость проекта, *Xgl* уже напичкан функциями больше, чем модный сотовый телефон прямо из Японии.



Xgl **размещает виртуальные рабочие столы на гранях куба (или восьмигранной призмы, если хотите), позволяя прокручивать его с помощью мыши.**

ВНАЧАЛЕ БЫЛ LINUX... XGL

Хотя настройка *Xgl* по умолчанию великолепна сама по себе, есть десятки параметров, с которыми можно поэкспериментировать, добившись абсолютной точности. Некоторые из них доступны в диалоге Desktop Effects в Центре управления Gnome (если вы используете SUSE 10.1 с нашего диска), но для Истинно Предельной Мощности потребуется *Gconf*. Вы можете запустить его, набрав **gconf-editor** в командной строке, или открыв Applications > System > Configuration > Gnome Configuration Editor.

Как только *Gconf* загрузится, откройте вкладку Apps и выберите Compiz. Там вы увидите древовидные структуры General и Plugins, в которые можно внести свои изменения. Не забывайте, что в Gnome любые изменения, сделанные в *Gconf*, применяются автоматически, без нажатия кнопки Save. Вот несколько ключей, с которыми вам, возможно, захочется поиграть:

- **general > allscreens > options > texture_filter** – установите в Fast, если у вас медленная видеокарта, или в Best в противном случае.
- **general > allscreens > screen0 > options > lighting** – когда эта опция включена, на ваш рабочий стол-куб падают лучи прямого света.
- **general > allscreens > screen0 > options > size** – число граней вашего трехмерного рабочего стола. По умолчанию оно равно 4 (это куб, поскольку верхняя и нижняя грани не задействованы), но вы можете установить что-нибудь до 32-х.
- **plugins > cube > screen0 > options > in** – включив эту опцию, вы окажетесь не снаружи вашего куба, а внутри. Внимание: эта опция не для подверженных клаустрофобии!
- **plugins > cube > screen0 > options > skydome** – укажите здесь файл PNG, находящийся на вашем компьютере, чтобы *Xgl* отображал изображение (видимое только во время его вращения) позади куба.
- **plugins > scale > screen0 > options > corners** – настройки эффекта Exposè. По умолчанию это TopLeft – если захотите, можете добавить более одного «горячего угла» (это угол дисплея, куда надо подвести мышь, чтобы произошла активация этого эффект). Другая возможность – изменить значение initiate, для задания определенной клавиши на клавиатуре.
- **plugins > switcher > screen0 > options > saturation** – здесь определяется цветность фона, когда вы нажимаете Alt-Tab. По умолчанию используется 100, но если вы измените его на 0, цвет фона обернется черно-белой градиентной шкалой.
- **plugins > water > screen0 > options > rain_delay** – установите это значение в 1 и нажмите Shift-F9, на экране возникнет настоящий муссон.
- **plugins > wobbly > screen0 > options > frection** – установите в единицу, чтобы ваши окна колыхались как студень (совершенно безумное зрелище, если распахнуть окно на весь экран).
- **plugins > zoom > screen0 > options > filter_linear** – активируйте ее, и *Xgl* будет сглаживать экран при увеличении масштаба, это выглядит намного приятнее!



Реформа Alt-Tab: встречайте живые миниатюры ваших окон, с альфа-прозрачностью для подсветки вашего выбора.

Эффекты Xgl

Что же это за функции? Одни имеют реальную ценность для конечного пользователя, другие – просто баловство; но впечатляюще выглядят все – и являются неотразимым аргументом против сидения сложа руки в ожидании Windows Vista.

• **Альфа-прозрачность и затухание**. Выберите любое окно, удерживая нажатой Alt, затем покрутите колесико мышки – прозрачность окна изменится. В некоторых других эффектах *Xgl* автоматически затемняет окна; этот же просто позволяет вам форсировать настройки.

• **Куб рабочих столов**. Виртуальные рабочие столы всегда с трудом воспринимались новичками в Linux, а теперь эту концепцию можно объяснить наглядно – разместив каждый из рабочих столов на кубе, который пользователь может вращать. Нажмите Ctrl-Alt и щелкните где-нибудь на экране, затем перетащите мышь влево или вправо, чтобы повернуть куб. Другой способ – щелкните на заголовке окна и перетащите его за пределы экрана, чтобы переместить его на другую сторону. Если вы нажмете Ctrl-Alt-Down («стрелка вниз»), *Xgl* покажет развертку куба, и вы увидите каждую грань; стрелками «Вправо» и «Влево» можно переключаться между гранями.

• **Экспозиция окон**. Вы видели это в OS X, теперь поглядите и в Linux: переместите вашу мышь в верхний левый угол вашего экрана, и *Xgl* сожмет все ваши окна так, чтобы они уместились на одном экране. Щелкните по любому из окон – оно будет выбрано; или щелкните по обоям, чтобы свернуть все окна.

• **Переключатель задач**. Нажмите Alt-Tab, и *Xgl* затемнит фон и все ваши окна, затем отобразит миниатюру каждого окна в центре экрана. При удержанной клавише Alt, щелкая Tab, вы сможете циклически переключаться между окнами на вашем рабочем столе, и *Xgl* выведет их на передний план с эффектом наплыва.

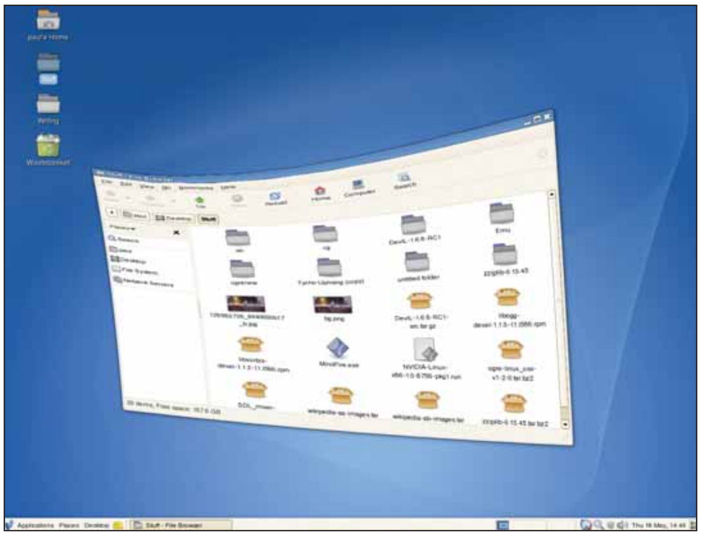
• **Вода**. Ну вот и один из бесполезных, но прикольных эффектов – нажмите Shift-F9, и на ваш рабочий стол польет дождь, создавая эффекты текучей ряби. При нажатии Ctrl-Windows капля воды упадет на указатель мыши, а удерживая Ctrl-Windows, можно нарисовать мышью свои собственные волны. Забавно, да только функции тут ни при чем!



Эффект струящейся воды демонстрирует гибкость и потенциал *Xgl*.

• **Трясущиеся окна**. Этот эффект, как и Вода, не особо полезен, но на экране смотрится просто изумительно. Когда вы перемещаете какое-нибудь окно (или увеличиваете его размер), окно реагирует так, как будто сделано из желе – край, который вы тянете, не отстает от курсора мыши, а остальная часть окна медлит следовать за ним, отчего слегка растягивается и деформируется.

• **Масштаб**. Удерживайте клавишу Windows и правую клавишу мыши, и *Xgl* перескочит на двукратное увеличение вашего рабочего стола. Прокрутите вперед колесико мыши – масштаб увеличится еще больше; и вы сможете перемещаться по рабочему столу, просто передвигая мышь.



Перетаскивание ваших окон временно их деформирует. Пользы никакой, но выглядит чертовски эффектно!



С одного щелчка ваши виртуальные рабочие столы выстроятся в одну линию, и вы сможете увидеть их все сразу. Перемещение влево и вправо – стрелками курсора.

Виджеты наготове



В Vista обещана новая боковая панель для управления виджетами – полезными утилитками (калькуляторы, часы, заметки-приклейки, RSS-ридеры и многие другие). Естественно, Linux уже долгие годы имеет все это в виде *SuperKaramba* – сверхгибкой системы виджетов с тысячами готовых тем, доступных для скачивания. Широкий выбор виджетов означает, что каждому что-нибудь да найдется – от удивительно полезного инструмента Liquid Weather (прогноз погоды в вашем районе на следующую неделю) до восхитительно нелепого монитора процессора *Doom*, отображающего загрузку процессора выражением лица героя из *Doom*. Когда нагрузка на систему возрастает, парень из *Doom* свирепеет на глазах. Возможно, Microsoft ему не устрашить, но есть дюжина-другая истинно полезных и бесплатных виджетов, обеспечивающих для Linux устойчивое лидерство.

Более того, уже ведутся работы по добавлению в KDE 4 совместимости с виджетами OS X Dashboard, что принесет еще сотни виджетов в каталог *SuperKaramba*.

Кстати, лучшее место для поиска виджетов – **www.kde-look.org**, там они представлены в соответствии с рейтингом и частотой скачиваний, тысячи штук на выбор. Но не хватите через край: слишком легко переборщить и замусорить свой рабочий стол.

ВНАЧАЛЕ БЫЛ LINUX... SUPERKARAMBA

Установив *SuperKaramba* на свою машину, обзаведясь новыми виджетами можно, всего-навсего нажав кнопку New Stuff в главном диалоге *SuperKaramba*. При этом загрузится список популярных виджетов, и вы сможете установить любой из них, выделив его и щелкнув Install. Чтобы отобразить понравившийся виджет, выделите его и щелкните на Add To Desktop.

По умолчанию местоположение каждого виджета зафиксировано, но вы можете щелкнуть на нем правой кнопкой мыши, выбрать Toggle Locked Position и переместить его вслед за мышью. Если виджетов явный перебор (а это может случиться очень легко, когда вы впервые дорветесь до этой «кондитерской» крутых наворотов), щелкните Close This Theme в контекстном меню виджета, который вам надоел.



В *SuperKaramba* есть виджеты на любой вкус, упрощающие жизнь.

Незамедлительная сеть

С Windows Vista установка сетевого соединения между разнообразными ПК и устройствами проста и интуитивно понятна? Хорошо, попробуем и это – и удержимся во главе гонки, даже не запыхавшись.



Установите флажок – и ваша музыка зазвучит в локальной сети.

Linux уже располагает простой сетевой функциональностью, использующей стандарт Apple Bonjour (ранее известный как Rendezvous или ZeroConf). Это сетевая широковещательная система, которая рассылает общее "hello" всем устройствам в сети и ждет ответа. Все другие устройства, на которых работает программа Bonjour (включая машины с Mac и Linux), в ответ сообщают о своем существовании, и устройства автоматически настраиваются для общения друг с другом.

Поскольку на самом деле Bonjour – просто протокол опроса сети, он может работать со всеми типами приложений. Сейчас наиболее популярное его применение – поиск общедоступной (shared) музыки в вашей сети. Если вы запустите *iTunes* на Mac или Windows (*iTunes* совместим с Bonjour), то сможете выбрать Share My Music (Поделюсь музыкой). Каждый, кто использует Linux-компьютер в этой сети, получит вашу музыку в свой проигрыватель и сможет слушать ваши мелодии.

Bonjour также популярен в мире обмена мгновенными сообщениями, поскольку позволяет людям формировать стихийные (*ad hoc*) коммуникационные сети. Например, если вы участник конференции, с терминала Wi-Fi вы можете подключиться и увидеть список других участников, подсоединенных к сети в настоящий момент. Затем начинайте общение через внутреннюю сеть – с использованием либо мгновенных сообщений, либо голосового или видео-чата.

Используя Linux-программу *iFolder*, вы можете предоставить свои файлы в общий доступ по сети другим людям, которых обнаружил Bonjour. Поскольку *iFolder* предоставляет общий доступ к папке на вашем компьютере, как если бы она была локальной, это означает, что если кто-то изменит какие-то файлы в этой папке, изменения будут автоматически скопированы всем, кто разделяет эту папку – превосходно для групповой работы в сети.

Прочная защита

Когда в октябре 2001 г. появилась Windows XP, Microsoft сулила пользователям «возросшую надежность, удобство, функции безопасности и коммуникации». В августе 2004 г. Microsoft выпустила «Windows XP Service Pack 2 с передовой технологией безопасности», предоставлявший «последние обновления безопасности и нововведения от Microsoft», «сильную безопасность по умолчанию» и «новые функции проактивной защиты».

Несмотря на столь напористую рекламу, многие люди до сих пор, похоже, страдают от массовых атак шпионских и рекламных программ и вирусов и вынуждены искать защиты у таких производителей, как McAfee и Symantec. Так что, пока мы тут в LXF радуемся появлению двухядерных процессоров, ибо это означает, что мы сможем выделить ядро для SETI@Home, наши коллеги из Windows-журналов мечтают о двухядерных машинах только потому, что смогут одно ядро отдать антивирусной программе.

Microsoft объявил, что «Windows Vista включает встроенные функции защиты, которые делают ее даже более защищенной, чем прежние клиентские операционные системы Windows». После чудес, обещанных на XP, не удивительно, что кое-кто уже воспринимает такие объявления как заевшую пластинку. Но сейчас Microsoft вводит «инновации» в управлении пользовательскими учетными записями и шифровании файловой системы.

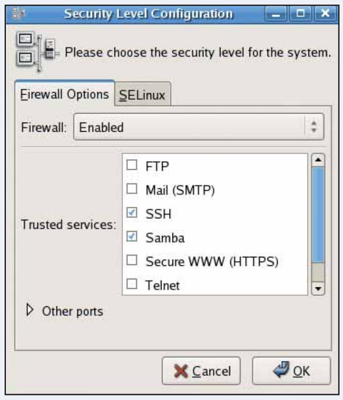
Как за каменной стеной

Многие машины с Windows XP позволяют любому подключенному пользователю устанавливать или удалять программы, или удалять файлы с жесткого диска. В случае Vista это сможет сделать только пользователь с правами администратора, а другие будут только запускать программы и работать со своими файлами. Разве не великолепная идея? Великолепная – причем настолько, что Linux использует ее с тех самых пор, как мелькнул огонек в глазах Линуса. Короче, Vista доблестно обзавелась системой пользователей в стиле Unix/Linux, чтобы наконец-то сделать машины Windows мало-мальски безопасными.

Шифрование файловой системы в Vista, названное BitLocker, также едва ли ново: Linux поддерживает шифрование томов уже несколько лет. Если вы хотите попробовать его прямо сейчас, воспользуйтесь пошаговым руководством из *LXF80* или просто отметьте флажок, как показано справа, во время установки вашего дистрибутива – вот так все просто.

Следующий большой скачок на пути к безопасности пользователей Windows: Vista представляет им брандмауэр. А разве брандмауэра не было в XP? И да, и нет: быть-то он был, но фильтровал только входящий трафик. Если в вашей XP есть уязвимость, которую можно атаковать удаленно, брандмауэр остановит хакеров и не даст ею воспользоваться. Но он не остановит исходящий трафик – шпионские и другие злонамеренные программы прекрасно могут «стучать» на вас без вашего ведома. Выходит, в Windows впервые появился приличный брандмауэр, блокирующий подозрительный трафик.

А вот Linux давным-давно имеет брандмауэр промышленного уровня, так что вполне простительно ваше недоумение: как же пользователи Windows столько лет жили без него?



Брандмауэр Linux – мощная броня, и настраивается он без проблем. Нам не дано понять, почему пользователи Windows только сейчас получают его как стандарт...

Vista? Баста!

Одну вещь мы пока не обсуждали: цену. Vista появится в январе 2007 года, и миллионы людей, получивших новый компьютер к Рождеству, неожиданно обнаружат, что «коробку» Windows надо еще купить – возможно, впервые. Привычно было иметь Windows предустановленным, так что «налог на Microsoft» не бросался в глаза. А тут пользователям действительно придется раскошелиться на обновление (в размере нескольких сот долларов) – и это серьезный расход. Но даже если продавать Vista по смешной цене, удастся ли покончить с современным настольным Linux? Многие из инноваций Vista уже доступны в Linux, зачастую работают быстрее (*Beagle*), выглядят приятнее (*Xgl*) или лучше соображают (Bonjour). Более того, можно ожидать, что мы увидим новые релизы SUSE, Fedora и Ubuntu, предоставляющие еще больше новых функций, в одно время с выходом Vista.



Рабочий Linux 2006: прочный как скала, с элегантной графикой и под завязку набитый программами для выполнения любых задач.

Ясно как день: людям, критиковавшим настольные системы Linux в прошлом, придется пересмотреть свои позиции в свете новых функций Vista, потому что Linux уже на световые годы впереди. Современная проблема Linux – отставание маркетинга: у Linux налицо все функции и простота использования, требуемые от высококлассной настольной системы, просто мы, нынешние линуксоиды, считаем это само собой разумеющимся.

Благодаря расширению рынка настольных систем Linux мы абсолютно уверены, что Vista будет последней Windows, выпущенной при столь большой доле рынка. Следующему релизу Windows – который, по-видимому, состоится около 2012 года – придется яростно сражаться, чтобы не стать просто последним. LXF

НЕ ЗАБУДЬТЕ, ЧТО...

Пользователи Linux давно привыкли ко множеству функций, аналоги которых только сейчас появляются в Vista. Просмотр во вкладках в *Internet Explorer*? Используйте *Firefox*. Векторный рендеринг графического интерфейса? В Gnome он появился несколько лет назад, благодаря *Cairo*. Автоматическое обновление ОС? Стар – в Linux это практикуется уже более пяти лет.

Есть масса функций Linux, которых в Vista нет вообще: файловая система, которая сама себя дефрагментирует без вмешательства пользователя (новички все не надивятся на этакий феномен); вирусоустойчивые почтовые клиенты; встроенная виртуализация; выбор окружения рабочего стола; и, конечно же, сотни гигабайт свободных программ, которые идут вместе с дистрибутивами.



Потрошим Gimp

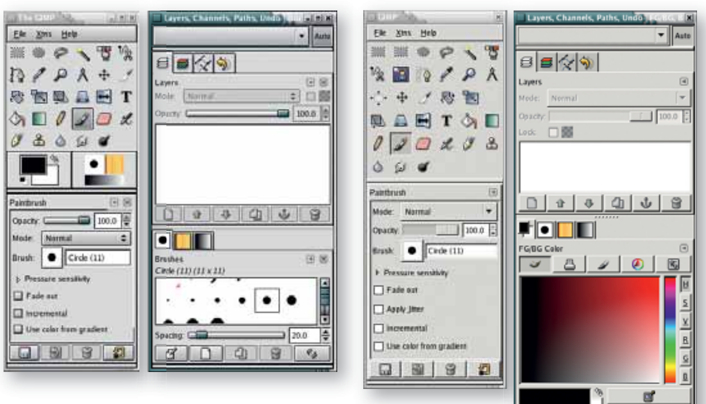
Есть много способов участвовать в разработке открытого проекта – например, помогать в написании документации или непосредственно создавать код. Почему бы не начать с Gimp? **Майкл Дж. Хэммел** проведет вас через весь процесс исправления ошибок, который он припас заранее...

НА ДИСКЕ

- *Gimp 2.2.11* и *2.3.8*
- Снимок *GECL CVS*

G*NU Image Manipulation Program* – более известный как *GIMP* – дедушка настольных приложений в мире открытых программ. Он вступил в жизнь как *Motif*-приложение в 1995 г. и привел к созданию *Gimp Toolkit* (также известного как GTK) и рабочего стола Gnome.

Хотя существуют настольные приложения и постарше, чем *Gimp*, ни одно из них не привлекло столько новых пользователей в мир Open Source или так повлияло на настольные системы. Но даже и проекты-дедушки нуждаются в постоянной поддержке открытого сообщества: разработчиков, писателей, художников, журналистов и пользователей.



Версии *Gimp*: текущая стабильная (слева) **и разрабатываемая** (справа)**. Обе имеются на диске, и вы можете их исследовать.**

Как помочь

Разработка *Gimp* – как и многих других открытых проектов – целиком зависит от информации от пользователей. Списки пожеланий и запросы на возможности посылаются через Bugzilla – это web-система, предназначенная для отслеживания статуса ошибок – и просматриваются командой разработчиков *Gimp*. Принятые пожелания, а также набор требуемых исправлений, перебираются в грядущий релиз. Пока разработчики вгрызаются в новые возможности, ошибки и изменения документации, пользователи тестируют выпущенные версии.

В конечном счете новый стабильный релиз становится доступным в виде исходных текстов. Поставщики Linux, такие, как Red Hat, Mandriva и Ubuntu, затем собирают новые версии и включают в свои дистрибутивы.

Разработка осуществляется двумя очень маленькими командами: основной командой и рядом участников. Основная команда фактически состоит всего из двух человек: Свена Ньюмена [Sven Neumann] и Митча Неттерера [Mitch Natterer]. *Ньюмен де факто* является лидером проекта, однако проект структурирован не жестко, поэтому важные решения часто принимаются только после серьезного изучения откликов остальных членов команды.

Активные участники – группа человек в 30, они работают над различными разделами проекта, включая исходный код, документацию и управление ресурсами типа репозитария *CVS*. Другие разработчики прошлого (включая автора этой статьи!) размещены на вкладке окна About.

Для помощи проекту вам не обязательно уметь кодировать, но вы должны быть хорошо знакомы с приложением *Gimp* с точки зрения конечного пользователя.

Ваша миссия

В этой статье мы продемонстрируем два лучших способа помочь проекту: охоту за ошибками и их исправление.

Охота за ошибками может быть случайной или целевой. Целевое тестирование ставит своей задачей изолировать проблему и описать ее более подробно –



Элита команды *Gimp*: ведущие разработчики Митч Неттерер (третий слева внизу) и Свен Ньюман (четвертый слева в среднем ряду).

РЕСУРСЫ УЧАСТНИКА

- **http://developer.gimp.org** Информация для разработчиков *Gimp*
- **www.gimp.org/bugs/howtos/bugzilla.html** Описание ошибок
- **www.gimp.org/bugs/howtos/submit-patch.html** О заплатках
- **www.le-hacker.org/papers/gobject** Объектная система *GLib*
- **http://developer.gimp.org/api/2.0/index.html** Документация по *API Gimp*
- О стилях написания кода см. файл **HACKING** в исходном коде.

Все это обязательно к прочтению. Однако если вы хотите реального взаимодействия с другими участниками, то надо подключиться к почтовым рассылкам и IRC-каналам:

- **www.gimp.org/mail_lists.html**
- Канал **#gimp** на **irc.gimp.org**

Существует много почтовых рассылок проекта. Основными являются рассылка для пользователей *Gimp* (по основным вопросам использования) и для разработчиков *Gimp* (по основным вопросам разработки). Живое обсуждение происходит на IRC-каналах. Более подробную информацию см. в wiki на **http://wiki.gimp.org/gimp/irc**.

это особенно актуально для трудновоспроизводимых ошибок. Охотники за ошибками должны быть очень хорошо знакомы со стабильной версией *Gimp*, а целенаправленные охотники – уметь компилировать новые версии программы. Познакомьтесь также с Bugzilla, мы представим ее вам на стр. 50.

Что касается исправления ошибок, то ядро *Gimp* написано на C, но есть и много дополнительных модулей на различных языках скриптов: самые популярные – Perl, Python и Script-Fu (вариация Scheme). Вы должны быть знакомы с созданием заплаток; инструкции можно найти на странице **www.Gimp.org**, но мы рассмотрим этот процесс более подробно на стр. 52.

Еще до начала...

Если вы собираетесь порыться в исходном коде, то прежде вам необходимо знать о нескольких инструментах.

CVS

Каждый участник *Gimp* знаком с *CVS*, программным обеспечением, которое управляет исходным кодом *Gimp*. Пользователи получают код из *CVS*, производят изменения, создают заплатку и отправляют ее команде разработчиков или участникам.

Bugzilla

Bugzilla – web-система, используемая для отслеживания изменений в проекте *Gimp*. Это могут быть сообщения об ошибках, запросы на новые возможности или изменения в документации. С первого раза Bugzilla немного ошеломляет, но команда *Gimp* усердно поработала, чтобы облегчить вхождение в свой мир. На странице *Gimp* вы найдете всю информацию об использовании Bugzilla для сообщений об ошибках и их поиске; а на странице разработчиков *Gimp* (*см. врезку Ресурсы участника*) содержатся более подробные указания для опытных пользователей Bugzilla, о том как отыскать уже имеющиеся сообщения.

C, Autoconf и другие инструменты

Исходный код ядра *Gimp* использует набор инструментов Autoconf (*Autoconf, Automake, Libtool*) для управления сборкой. Знание Autoconf желательно, но не обязательно. Зато важно умение программировать на C – оно требуется для работы с ядром *Gimp* и многими модулями. Для некоторых модулей необходимо знать Perl, Python и Script-Fu, но для работы с кодом ядра эти языки не потребуются. Пригодится знакомство с текстовым редактором и такими инструментами, как *Cscope*, *Strace*, *GDB*. Надо также вникнуть в стиль программирования *Gimp* (*снова, см. врезку Ресурсы участника*).

Не программист?

Прежде чем начать охоту за ошибками, кратко опишем другие способы помочь проекту *Gimp*.

Первый – поддержка пользователей. Команда разработчиков проводит большую часть своего свободного времени за работой над исходным кодом, поэтому у них мало времени на ответы новичкам. Достаточно освоившись с программой, вы можете обеспечивать поддержку конечных пользователей посредством списков рассылок, web-страниц и чатов.

Такая помощь очень важна: ваша поддержка – видимая часть проекта, обращенная к публике. Поэтому обязательно надо полностью понимать, как пользоваться программой, и вежливо реагировать даже на самую жесткую критику.

Также очень ценны переводы и документация. В переводе нуждается как документация, так и интерфейс программы. Это прекрасная возможность помочь проекту: работа над большей частью документации требует лишь знания какого-либо текстового редактора, а работа над web-сайтами – знания XML и формата DocBook.

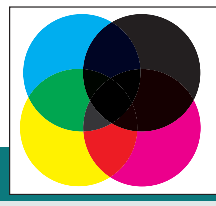


***Gimp* содержит много проектов, под управлением системы контроля исходного кода *CVS*.**

Опытные программисты могут попробовать задуматься о расширении ключевых возможностей. Общеизвестный пример – поддержка 16-битовых цветовых каналов. Потребность в ней назрела давно, однако ее реализация потребовала неожиданно большого объема работ. В итоге запланирован пересмотр исходного кода *Gimp*: он будет использовать код вспомогательного проекта под названием GEGL (*см. врезку внизу*).

В какой форме ни выразится ваше участие, *Gimp* представляет собой отличный старт, и мы надеемся, что опыт этой захватывающей работы послужит вам хорошей наградой. Прежде чем нырнуть в исходный код, взгляните на врезку Ресурсы участника, расположенную вверху. Если мы вдохновили вас на поиск и исправление ошибок, можете начинать – просто переверните страничку!

>>



GEGL: БУДУЩЕЕ GIMP

GEGL означает *Generic Graphical Library*; де факто это будущее проекта *Gimp*. Пока проект находится в процессе разработки, а когда будет закончен, станет внутренней библиотекой обработки в *Gimp*. Кроме поддержки 16-битовых цветовых каналов (одна из первоначальных целей), *GEGL* предоставит богатую функциональность для обработки изображений, включая управление цветом, редактирование цветовых пространств CMYK (см. картинку вверху справа) и L*a*b, гибкие системы на основе мозаики и многопоточную обработку изображений. Также он предоставит механизм на основе направленного графа для запоминания серии манипуляций над изображением.

Большинство этих возможностей требуется профессиональным пользователям – видеоредакторам, например. Но и средний пользователь также найдет их полезными. Так, механизм на основе направленного графа позволяет хранить сложную последовательность шагов при создании изображения, от добавления текста до применения фильтров, и даже редактировать эти шаги. Допустим, вы изготовили в *Gimp* логотип для web-странички и сохранили ваши действия в файле. Позже вы можете загрузить файл и создать, например, логотип для рекламного плаката – указав другое разрешение или размер холста.

Официальной даты вхождения *GEGL* в состав *Gimp* не существует, но есть надежда увидеть его в очередной главной версии после *Gimp 2.4*. А до тех пор проект нуждается в разработчиках и тестировщиках – и вы знаете, что делать.

www.gegl.org



« Часть 1 Выходим на охоту

Прежде чем исправлять ошибки, надо сначала о них узнать – поэтому заглянем в Bugzilla!

Лучший способ понять, что такое работа над проектом – это исправить в нем ошибку.

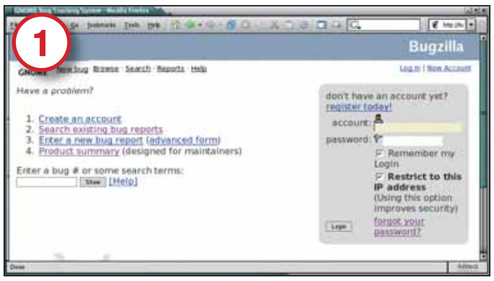
С точки зрения вклада в проект *Gimp*, ошибкой считается все, что существует в исходном коде *Gimp* и работает не так, как должно – в отличие от запроса на возможность, для которой исходного кода либо нет вообще, либо он есть, но спроектирован нежелательным образом.

Если ошибка обнаружена, очень важно суметь ее снова воспроизвести, тут-то и пригодится ваша помощь: просто играйте с программой, пока не сведете к минимуму число факторов, вызывающих ошибку. Затем опишите их в виде последовательности шагов, чтобы любой желающий смог прочесть ваше сообщение в Bugzilla.

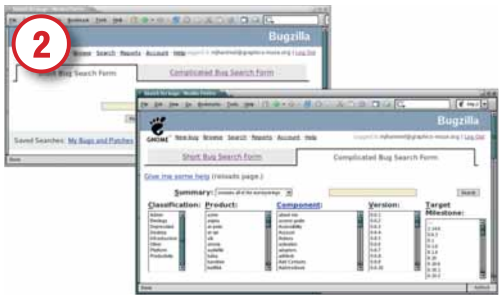
База Bugzilla (**http://bugzilla.gnome.org**) служит для многих проектов, связанных с рабочей средой Gnome. Прежде чем искать в ней сообщения об ошибках, вам понадобится зарегистрироваться и получить ID.

Выбрав ошибку по вкусу, скачайте пос-

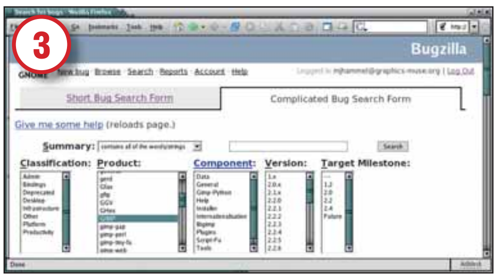
КАК НАЙТИ СООБЩЕНИЯ ОБ ОШИБКАХ В BUGZILLA



Войдите в базу данных Bugzilla и нажмите на ссылку Search вверху страницы, затем перейдите к Complicated Bug Search Form – Запросу на расширенный поиск. Для поиска ошибок воспользуемся специальным ключевым словом gnome-love, им команда разработчиков *Gimp* помечает легко устранимые ошибки, чтобы их можно было легко найти.



При поиске сообщений об ошибках надо обратить внимание на два момента. Первый –список продуктов. Листайте его до тех пор, пока не наткнетесь на запись Gimp. Нажмите на нее. Произойдет обновление других списков, но мы пока их проигнорируем.



Прокрутите экран до области, означенной как Advanced Searching Using Boolean Charts – Продвинутый Булев поиск. Там есть два выпадающих списка. В первом списке выберите Keywords (Ключевые слова), а во втором – Contains The String (Содержит строку). В текстовом поле рядом со списком наберите gnome-love. Поиск подготовлен.

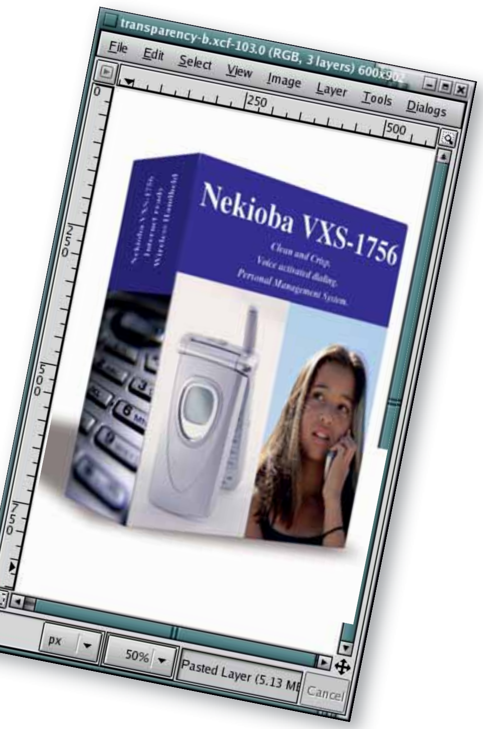


Нажмите кнопку Search (она находится прямо над списками, которые вы только что модифицировали). Просмотрите колонку Summary – если какая-либо ошибка вам приглянется, нажмите на ее номер, чтобы узнать о ней больше (*см. таблицу ниже*). Мы выбрали ошибку #331839.

5

ID	Sev	Pri	OS	Product	Status	Resolution	Summary
331839	enh	Nor	All	Gimp	NEW		Clear keyboard shortcuts

Данная таблица информирует о выбранной ошибке Gimp. Во втором столбце (Степень серьезности ошибки) стоит enh, сокращение от enhancement (улучшение), т.е. формально это не ошибка, а запрос на улучшение. (Таблицу заполняют разработчики, получив через Bugzilla извещение об ошибке.) В данном случае решено изменить диалог Preferences: добавить возможность очистить клавиши быстрого доступа без их сброса в первоначальное состояние. Мы думаем, что сможем с этим справиться. Вот и попробуем.



леднюю копию *Gimp* из *CVS*. Чтобы она заработала, понадобятся следующие зависимости:

- *autoconf 2.54* или старше.
- *automake 1.7* или старше.
- *libtool 1.4* или старше.
- *gettext 0.13* или старше.
- *GTK 2.8.10* или старше.

Возможно, первые четыре пакета у вас есть, но бьемся об заклад, что последнего нету. *GTK* имеет свои зависимости. Установить их легко, но надо позаботиться, чтобы они ничего не сломали в вашей системе.

Итак, скачайте и распакуйте *GTK*. В любом современном дистрибутиве вам скорее всего понадобятся последние *GTK*, *Glib*, *ATK*, *Pango* и Cairo. Первые четыре доступны на сайте *GTK* (**www.gtk.org/download**). Архив Cairo также содержится на сайте *GTK*, но на другой странице (**ftp://ftp.gtkK.org/pub/gtk/v2.8/dependencies**). Важно собрать пакеты в правильном порядке: Cairo, *GLib*, *Pango*, *ATK*, *GTK*.

Перед сборкой скажем каждому пакету, где искать библиотеки и информацию о конфигурации:

```
export PKG_CONFIG_PATH=/usr/
    local/gtk+-2.8/lib/pkgconfig:$PKG_
    CONFIG_PATH
export LD_LIBRARY_PATH=/usr/
    local/gtk+-2.8/lib:$LD_LIBRARY_
    PATH
```

Установите эти переменные среды, иначе в сборку попадут старые версии, уже установленные на вашей системе. В директории каждого пакета наберите команды:

```
./configure --prefix=/usr/local/gtk+-
    2.8
make
sudo make install
```

Мы тем самым указываем всем пакетам устанавливаться в **/usr/local/GTK+-2.8**. Благодаря этому новые версии ПО не будут мешать старым.

Сборка

Установив зависимости *GTK*, соберите *Gimp*. Необходимо зайти на *CVS*-сервер и скачать исходный код. Направьте вашу машину на *CVS*-сервер *Gimp*:

```
export CVSROOT=':pserver:
    anonymous@anonCVS.Gimp.org:/cvs/
    gnome'
```

и зайдите:

```
cvs login
```

Вы получите примерно следующее сообщение:

```
'Logging in to :pserver:
anonymous@anonCVS.Gimp.
org:2401/CVS/gnome
CVS password:'
```

Мы заходим как анонимный пользователь, поэтому у нас будут права только на чтение. Просто нажмите Enter (пароль вводить не надо).

Теперь скачайте исходный код *Gimp*:

```
cvs -z3 checkout Gimp
```

Флаг **-z3** говорит об использовании сжатия, для ускорения передачи файлов [и экономии трафика, – прим.ред.].

Готово? В вашем текущем каталоге должен появиться каталог **gimp**. Чтобы собрать код, наберите:

```
cd Gimp
./autogen.sh --prefix=/usr/local/
Gimp-2.3
make
sudo make install
```

Скрипт **autogen.sh** похож на скрипт *configure*, который вы запускали для *GTK*-приложений, и теперь вы передаете опцию **--prefix** скрипту **autogen.sh**, а не configure (заметим, что мы ставим *Gimp* в директорию, отличную от *GTK*). Переменные **PKG_CONFIG_PATH** и **LD_LIBRARY_PATH**, установленные перед сборкой *GTK*, укажут системе сборки *Gimp*, где искать *GTK*. Мы хотим поставить *Gimp* в отдельный каталог, чтобы потом мы смогли его удалить и собрать другую версию, не повредив *GTK*.

План атаки

Теперь мы можем осмотреть самую последнюю версию диалога Preferences и составить план, как исправить наш досадный недочет. Интересующая нас область обведена и выделена синим цветом на рисунке (*справа вверху*) – это опции Keyboard Shortcuts. В сообщении говорилось, что требуется опция очистки горячих клавиш и что существующие кнопки необходимо изменить на меню опций, подобное меню Navigation Preview Size.

Очень хорошо... Поищем код, ответственный за создание этой части диалога Preferences. В исходном коде *Gimp* находится много каталогов, включая:

- **app** код ядра *Gimp*.
- **data** Кисти, градиенты и прочее.
- **plugins** Фильтры и другие дополнительные модули.
- **po** Переводы текстов в интерфейсе пользователя, например, в меню.

Нам требуется через каталог **app** добраться до каталога **dialogs**, а в нем найти файл **preferences-dialog.c**. В файле поищите строку, которая находится на кнопке Reset,. Таким образом мы найдем код, с помощью которого создается эта кнопка:

```
button2 = prefs_button_add (
Gimp_STOCK_RESET,
_("_Reset Saved Keyboard Shortcuts
    to "
"Default Values"),
GTK_BOX (vbox2));
```



Вот что мы хотим изменить: кнопки, выделенные синим цветом, надо переделать на список опций. Думаете, справимся? Конечно, справимся!

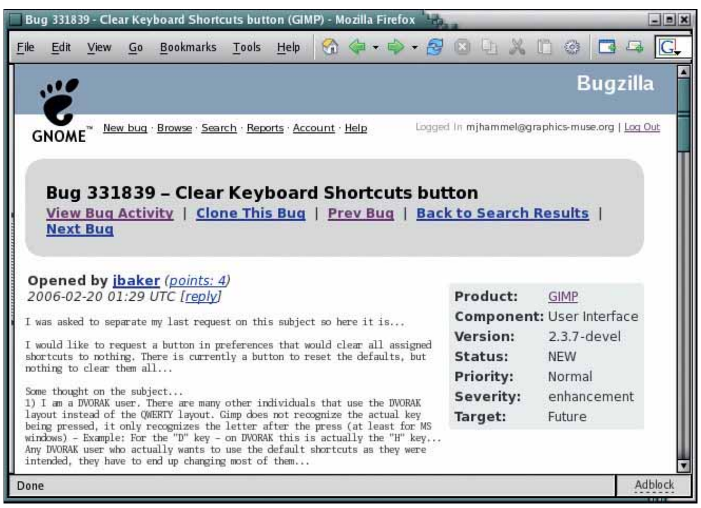
```
g_signal_connect (
button2, "clicked",
G_CALLBACK (prefs_menus_clear_
    callback),
Gimp);
g_object_set_data (
G_OBJECT (button),
"clear-button", button2);
```

Ага, вызовы **prefs_button_add()** надо заменить на один вызов функции **gimp_int_combo_box_new()**, создающей виджет меню – ее можно найти в документации по API *Gimp* (**http://developer.Gimp.org/api/2.0/libGimpwidgets/GimpIntComboBox.html**). Кроме того, обратные вызовы для обеих кнопок надо слить в один обратный вызов.

Перечень мероприятий готов; приступим же к его выполнению.



>>



Сообщение об ошибке может послать любой – только опишите ее поподробнее.

<< Часть 2 Поправки и заплатки

Задача поставлена – давайте выполним ее и отошлем результат команде разработчиков на проверку....

После трудов по локализации ошибки исправление оказывается простейшей частью. Мы описали его здесь, и когда вы будете исправлять другой код, шаги, которые мы предпримем, вам помогут.

Исправление

Добавим поле со списком прямо под существующими кнопками в **app/dialogs/preferences-dialog.c**. Чтобы пояснить, зачем это поле нужно, надпишем его (слева). Все это требует *GTK*-виджета 'hbox':

```
hbox = gtk_hbox_new (FALSE, 6);
gtk_box_pack_start (GTK_BOX
    (vbox2), hbox, FALSE, FALSE, 0);
gtk_widget_show (hbox);
label = gtk_label_new (_("Keyboard
    Shortcut Status:"));
gtk_box_pack_start (GTK_BOX
    (hbox), label, FALSE, FALSE, 0);
gtk_widget_show (label);
```

Затем добавим поле со списком из четырех записей. Первая запись – «пустышка»: она позволит сбросить меню после каждого действия, чтобы пользователь в любой момент смог перезапустить то же самое действие.

```
combo = Gimp_int_combo_box_new
    (
_("Choose an Action"),
    KBS_IGNORE,
_("Save Now"),
    KBS_SAVE,
_("Reset to Default Values"),
    KBS_RESET,
_("Remove All Keyboard Shortcuts"),
    KBS_CLEAR,
    NULL);
Gimp_int_combo_box_set_active
    (Gimp_INT_COMBO_BOX (combo),
    KBS_SAVE);
gtk_box_pack_start (GTK_BOX
(hbox), combo, FALSE, FALSE, 0);
    gtk_widget_show (combo);
g_signal_connect (combo,
    "changed",
G_CALLBACK (prefs_menus_
    keyboard_shortcuts),
Gimp)
```

Строка **g_signal_connect()** сообщает программе, что когда пользователь изменяет наше новое поле со списком, надо вызвать функцию **prefs_menus_keyboard_shortcuts()**. Эта функция получает ID виджета поля со списком (ID используется для обнаружения, какой пункт меню был выбран) и структуру **gimp**.

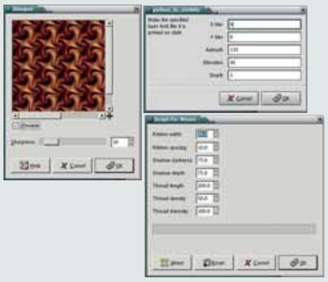
Каждый пункт меню в вызове **gimp_int_combo_box_new()** описан рядом с определяемым значением (**KBS_SAVE** и т.д.). Объединим их в перечисляемый тип и добавим его в начале файла:

```
enum {
KBS_IGNORE,
KBS_SAVE,
KBS_RESET,
KBS_CLEAR
};
```

Теперь надо заменить обратные вызовы функций для существующих кнопок одной функцией. Любые изменения в поле со списком должны осуществлять функции обратного вызова. Это просто. Новая функция будет просматривать поле со списком, определять, какой пункт меню был выбран, и использовать оператор **switch** для выбора соответствующего действия. В двух случаях (**KBS_SAVE** и **KBS_CLEAR**) действие останется как у прежних кнопок. Переключаясь на первый, пустой пункт меню, не будем делать ничего. Последний пункт «чистит» горячие клавиши: проходится по списку и сбрасывает их. После этого новый список требуется сохранить – и он будет доступен в следующий раз, когда пользователь запустит программу.

```
static void
prefs_menus_keyboard_
shortcuts(GtkWidget *combo,
Gimp *Gimp)
{
gint value;
gint fd;
GError *error = NULL;
gchar *filename;
Gimp_int_combo_box_get_active
(Gimp_INT_COMBO_BOX (combo),
&value);
switch(value)
```

ПИШЕМ ДОПОЛНИТЕЛЬНЫЙ МОДУЛЬ К GIMP



Модули на C, Python и Script-Fu: окна предпросмотра доступны только для написанных на C.

Дополнительные модули – это небольшие программки, написанные на каком-либо языке программирования и расширяющие возможности *Gimp*. Их часто называют фильтрами, просто потому, что большинство из них находится в меню Filters. Однако модуль может вставить себя в любой пункт меню и осуществлять не только обработку изображений, но также генерировать логотипы, шаблоны, обеспечивать ввод-вывод и даже действовать как сервер для внешних программ, чтобы отображать изображения через *Gimp*.

Модуль можно написать на C с использованием библиотеки *libgimp*, или же на одном из поддерживаемых скриптовых языков: Script-Fu (подмножество языка Scheme) и Python. Perl тоже поддерживается, но только как добавочный пакет. Каждый из скриптовых языков связан с API libgimp.

Чтобы написать модуль, нужно быть знакомым с PDB, Procedural Database [База Процедур], содержащей набор всех функций, которые можно вызывать из модуля. Просмотр PDB через осуществляется браузером Procedure Browser (Xtns > Procedure Browser) из меню в Toolbox. Каждую функцию можно искать по имени, причем предоставляется краткое описание параметров. Способ вызова функций зависит от языка, на котором написан модуль. В C вы можете сделать следующее:

```
image_id = gimp_image_
new(width, height, type);
```

тогда как в Perl это может выглядеть так:

```
image_id = $image->new(width,
height, type);
```

а на Python вот так:

```
image_id = gimp.image(width,
height, type)
```

Хороших руководств по написанию модулей немало. Для написания на C смотрите серию руководств из трех частей на сайте разработки *Gimp*, по адресу **http://developer.gimp.org/plug-ins.html**. Отличное руководство для языка Python доступно в сети на **www.gimp.org/docs/python/index.html**.

Документация для Perl входит в состав пакета **Gimp-Perl** – ее можно прочесть, набрав perldoc **Gimp** или посетив страницу **www.gimp.org/tutorials/Basic_Perl**. Желаем удачи!





Наша обновленная панель Keyboard Shortcuts в диалоге Preferences; новое меню отображается в правой нижней части окна.

```
{
case KBS_IGNORE:
break;
case KBS_SAVE:
menus_save (Gimp, TRUE);
g_message (_("Your keyboard
shortcuts have been saved."));
break;
case KBS_RESET:
menus_clear (Gimp, &error);
g_message (_("Your keyboard
shortcuts will be reset to default "
"values the next time you start
Gimp."));
break;
case KBS_CLEAR:
menus_remove (Gimp);
menus_save (Gimp, TRUE);
g_message (_("Your keyboard
shortcuts will be cleared "
"the next time you start Gimp."));
break;
}
Gimp_int_combo_box_set_active
(Gimp_INT_COMBO_BOX (combo),
KBS_IGNORE);
}
```

Мы добавили две новых записи в файл **app/menus/menus.c** и добавили функцию **menus_remove()**, вызываемую из **prefs_menus_keyboard_shortcuts()**. Она производит сброс всех горячих клавиш в цикле.

```
void
menus_remove (Gimp *Gimp)
{
gtk_accel_map_foreach(0,menus_
remove_actions);
}
static void
menus_remove_actions (gpointer
data,
const gchar *accel_path,
guint accel_key,
GdkModifierType accel_mods,
gboolean changed)
{
gtk_accel_map_change_entry
(accel_path, 0, 0, TRUE);
}
```

Нужно также написать прототип функции **menus_remove()** в заголовочном файле **app/menus/menus.h**, чтобы ее можно было вызывать из **preferences-dialog.c**.

Проводим тестирование

Тяжелая работа позади. Выполним перекомпиляцию и переустановку – теперь можно запустить приложение и проверить работу изменений.

```
make
sudo make install
/usr/local/Gimp-2.3/bin/Gimp-2.3
```

Простейший способ проверить изменения – просмотреть в меню File. Прежде чем запускать новую опцию Remove All Keyboard Shortcuts, обратите внимание, что к пунктам меню File и Save приписаны горячие клавиши. Когда мы запускаем новую опцию, эти клавиши исчезают. Выйдем из программы и запустим ее снова – увидим, что клавиш нет. Если затем в меню выбрать пункт Reset, выйти и перезапустить *Gimp*, все горячие клавиши будут восстановлены по умолчанию.

Создание заплатки

Изменения, значит, работают. Время создать заплатку и послать ее на суд почтенным разработчикам *Gimp* – так вы будете поступать и при самостоятельной работе над исправлениями. Из корня каталога исходного дерева *Gimp* наберите

```
cvs diff -up > ../patchfile.patch
```

В каталоге, расположенном на один уровень выше, чем текущий, в котором вы работаете, создастся файл заплатки. Команда *diff* может проработать довольно долго. Проверьте, что файл содержит только изменения, сделанные вами: команда способна протащить туда и постороннюю информацию.

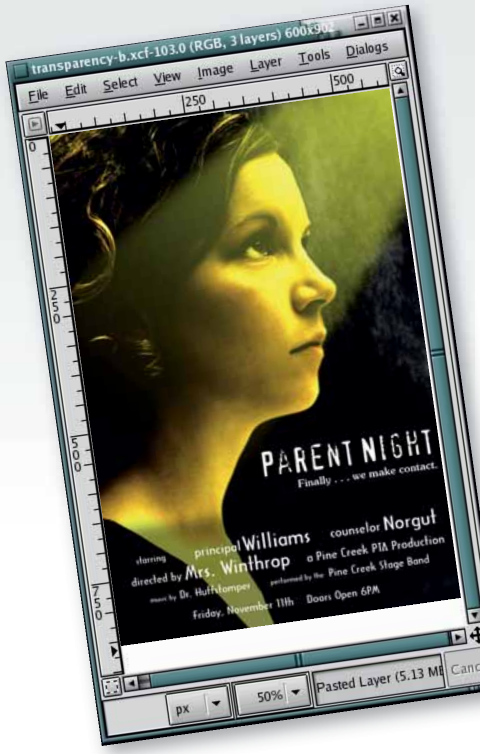
Внизу сообщения об ошибке в Bugzilla есть поле, содержащее информацию о проделанной работе. Заполните секцию Комментарий и присоедините заплатку с помощью ссылки, расположенной ниже. Задача выполнена: можете самодовольно откинуться в кресле.

И последнее. Разработчики ответили, что наша реализация будет работать лучше, если вместо меню будет задействована третья кнопка. Так что мы опять взялись за дело и написали код для третьей кнопки. После этого Митч Неттерер заплатку принял.

Теперь вы поняли: процесс абсолютно прямолинейный, поэтому изучайте, участвуйте и создавайте! LXF

БЛАГОДАРНОСТИ

Эта статья основана на слайдах, представленных Карин Делвер [Karine Delvare] на конференции Libre Graphics, прошедшей ранее в этом году. Дополнительную помощь оказали Кэрол Спирс [Carol Spears], Митч Неттерер [Mitch Natterer] и Уильям Скэггс [William Skaggs]. Особая благодарность Карин Делвер (и ее мужу) и Кэрол Спирс за предоставленные фотографии разработчиков на встрече Libre Graphics.



КАКОВА СУДЬБА ИСПРАВЛЕНИЯ LXF?

В нашем исправлении кнопки располагались друг рядом с другом. Но из-за ограничений на ширину текста разработчики решили добавить еще одну кнопку пониже исходных. Обычному пользователю это может не пригодиться никогда, но продвинутые пользователи, любители персональных конфигураций, оценят новую возможность.

В будущем можно добавить варианты предустановок горячих клавиш: выбор конфигурации по умолчанию, как в *Photoshop* или как в Mac OS.





До (слева) **и после** (справа)**: в диалог Preferences добавилась новая опция. Продвинутым пользователям она очень понравится.**

IBM WEBSPHERE COMMUNITY EDITION CONTEST 2006

Компания LinuxCenter.Ru при поддержке корпорации IBM представляет вашему вниманию конкурс **IBM WebSphere Contest 2006**! Это мероприятие нацелено на разработчиков, инструкторов, а также всех, кому интересны технологии Java и Open Source.

Что нужно для участия? В первую очередь – установить в своей системе IBM WebSphere Application Server Community Edition (WAS CE). Если вы регулярно читаете наш журнал, то уже наверняка знакомы с этим открытым сертифицированным сервером J2EE-приложений, базирующемся на Apache Geronimo. Если же вы по каким-то причинам пропустили апрельский номер – не отчаивайтесь: всю необходимую для быстрого старта информацию можно найти по адресу http://www.linuxformat.ru/contest/was2006.phtml#quickstart. Дистрибутив IBM WebSphere Application Server Community Edition можно найти на прилагаемом к LXF78 диске: Сторона 2: Websphere CE/kick-start_ocd.iso или бесплатно загрузить отсюда: http://www.ibm.com/developerworks/downloads/ws/wasce/?S_TACT=105AGX28&S_CMP=DLMAIN. Перед установкой сервера данный ISO-образ необходимо предварительно записать на CD.

Основная задача конкурса – популяризация IBM WebSphere Application Server Community Edition/Apache Geronimo как открытого и эффективного средства для решения широкого круга различных задач, поэтому мы приветствуем не только готовые к использованию приложения, но и инструменты для интеграции WAS CE с другим свободным ПО (в первую очередь, Eclipse), проверки компонентов, шаблоны приложений WAS CE/Geronimo и так далее. Конечно, ваш творческий полет не ограничивается написанием кода – мы будем рады видеть методические материалы: серьезные статьи, планы учебных курсов, технические задания для учебного проекта и так далее.

Победители конкурса будут определены Экспертным советом, в который войдут представители ведущих софтверных компаний. Подведение итогов конкурса и торжественная церемония награждения победителей пройдет на выставке LinuxLand/SofTool'06 (Москва, ВВЦ, 26-29 сентября 2006 года). Победители конкурса в каждой из номинаций получат призы, предоставленные компанией LinuxCenter.ru:

- ноутбук
- программное обеспечение, литература и атрибутика от LinuxCenter.Ru
- подписка на журнал Linux Format от редакции журнала.

Статьи и другие методические работы лауреатов конкурса будут опубликованы на сайте IBM developerWorks и на страницах генерального информационного спонсора – журнала Linux Format.

Впереди – целое лето, а как показывает опыт программы Google Summer of Code, за это время можно успеть многое. Желаем удачи в ваших начинаниях!

WAS CE – РУКОВОДСТВО К БЫСТРОМУ СТАРТУ

Для того, чтобы начать работу с текущей версией WAS CE (прежде всего, запустить этот сервер), необходимо:

>> Использовать одну из следующих операционных систем:

- Red Hat Enterprise Linux 3/4
- SUSE Linux Enterprise Server v9
- Windows 2003
- Windows XP SP2

Это те системы, на которых WAS CE прошел сертификацию на соответствие спецификациям J2EE 1.4, хотя сервер может работать и на других платформах.

>> Установить JDK 1.4.2_9 и старше, но не JDK 1.5.

Для запуска готовых приложений достаточно иметь Java Runtime Environemnt (JRE). На компакт-диске Java Kick-start находится JRE 1.4.2 от IBM, но можно использовать и JRE от Sun Microsystems.

Для разработки приложений необходимо наличие JDK 1.4.2 любого из вышеназванных производителей. Sun JDK (J2SE 1.4.2) можно бесплатно загрузить по адресу

http://java.sun.com/javase/downloads/index.html

После установки JDK имеет смысл добавить в переменную окружения **PATH** каталог **<jdk_install_dir>/bin**.

Хотя это и не относится непосредственно к WAS CE, многие Java-приложения используют переменную окружения JAVA_HOME. Ей можно присвоить значение в виде имени каталога установки JDK.

Теперь можно приступать к установке сервера с диска Java Kick-start.

Последнюю версию WAS CE можно также загрузить с сайта IBM по адресу:

http://www.ibm.com/developerworks/downloads/ws/wasce/?S_TACT=105AGX28&S_CMP=DLMAIN

Разработчики могут обращаться к документации как по WAS CE (входит в состав дистрибутива), так и по Geronimo. Документация по Geronimo доступна по адресу

http://geronimo.apache.org/.

Статьи на русском языке, объясняющие различные аспекты установки и использования Geronimo, можно найти также по адресу:

http://www-128.ibm.com/developerworks/ru/



УСЛОВИЯ КОНКУРСА

ОБЩИЕ ПОЛОЖЕНИЯ

1. Участвовать в конкурсе «IBM WebSphere Contest 2006» может любой человек, ознакомившийся с настоящими условиями и согласный с ними, за исключением сотрудников IBM, LinuxCenter.Ru, журнала Linux Format и членов их семей.
2. Учредителями конкурса выступают корпорация IBM и компания LinuxCenter.Ru, генеральным информационным спонсором – журнал Linux Format.
3. Конкурс проводится в период с 10 мая по 10 сентября 2006 года. Поведение итогов конкурса и церемония награждения победителей состоится в ходе выставки LinuxLand/Softool'2006 (Москва, ВВЦ, 26-29 сентября 2006 года). Итоги конкурса будут также опубликованы в ноябрьском номере журнала Linux Format.
4. Победители конкурса в каждой из номинация определяются Экспертной комиссией, состав которой утверждается Учредителями конкурса. Решение Экспертной комиссии является окончательным. Денежный эквивалент призов не выплачивается.

ПОРЯДОК ПРЕДСТАВЛЕНИЯ РАБОТ

1. Присланные на конкурс работы должны попадать в одну из трех номинаций:

   I. Методическое и информационное обеспечение.

   Это может быть большая, серьезная статья с детальным рассмотрением конкретных особенностей WAS CE версии 1.0 и выше; план учебного курса с учебными примерами и системой контроля усвояемости материала; грамотное техническое задание для учебного проекта: проработанная структура сайта на эту тему и т.п.

   II. Инструментарий

   Это могут быть расширения для Eclipse, облегчающие работу с WAS CE, утилиты для проверки компонентов, перекодировки дескрипторов для перехода на WAS CE с других J2EE-серверов, шаблоны (templates) для разработки приложений с использованием WAS CE.

   III. Решения.

   Это готовые приложения, компоненты и любые законченные и готовые для использования фрагменты кода, устанавливаемые на серверах WAS CE/Geronimo.

2. Имущественные авторские права на работы, представленные в номинации **I**, должны принадлежать участнику конкурса. Это подразумевает, что статья или любая другая методическая разработка не может быть опубликована в печатных или сетевых изданиях (журналах, сборниках, сайтах и т.п.), ограничивающих право автора на размещение материала в других источниках. Код, содержащийся в статьях или методических разработках, должен быть представлен в форме, удобной для проверки его работоспособности и эффективности, то есть в виде полных исходных текстов и сценариев для сборки.
3. Работы, представленные в номинациях **II** и **III**, должны распространяться на условиях, одобренных Open Source Initiative, то есть по какой-либо открытой лицензии. Распространение кода на тех же условиях, что и WAS CE/Geronimo – приветствуется.
4. Конкурсные работы следует высылать на адрес электронной почты: contest@linuxformat.ru Допускается (а в случае с работами, попадающими в номинации **II** и **III** – приветствуется) публикация конкурсных материалах на web-сайтах, не ограничивающих имущественные права автора (SourceForge.net, домашние страницы авторов и т.п.). В этом случае на адрес contest@linuxformat.ru следует выслать лишь ссылку на такой сайт с кратким описанием представляемой работы.

ПОРЯДОК КОНКУРСНОГО ОТБОРА

1. Представленные на конкурс работы будут оцениваться по следующим критериям:

   - актуальность и востребованность (в том числе потенциальная – по мнению экспертов);
   - качество решения и его оформления, включая документацию;
   - степень универсальности решения.




ИЛЛЮСТРАЦИЯ: STUART HARRISON

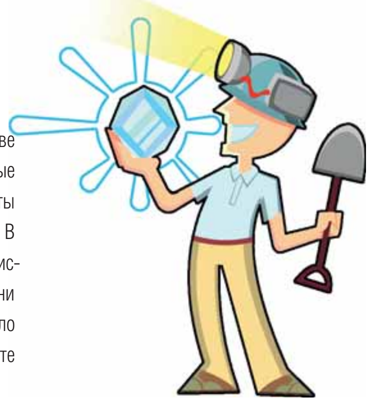
НА ДИСКЕ

ТАЙНЫЕ КЛАДЫ

Сокровищницы приложений с открытым кодом ждут своего открытия. Поохотьтесь за ними вместе с **Майком Сондерсом**.

Надеемся, вам понравятся наши десять тайных кладов. Если вы хотите поделиться своими находками, дайте нам знать **letters@linuxformat.ru**.

Огромный каталог приложений Freshmeat содержит более 40 000 проектов, и почти все они – Linux-приложения с открытым кодом; но все ли они стоят внимания? Издатели дистрибутивов собирают сливки этого изобилия и размещают их на своих дисках или в репозиториях; но при огромном количестве разрабатываемых приложений некоторые прекрасные (но малоизвестные) проекты неизбежно остаются незамеченными. В этом месяце мы просеяли Интернет в поиске классных приложений, о которых вы ни разу не слышали, или вам некогда было их опробовать. И, конечно же, вы найдете их на DVD!!



RSS-GLX



Удивите друзей роскошными хранителями экрана

ИНФОРМАЦИЯ

ЧТО: Коллекция хранителей экрана
ГДЕ: http://rss-glx.sourceforge.net
ЗАЧЕМ: Разнообразьте свой рабочий стол.

Устанавливаемые версии Gnome и KDE содержат стандартный набор хранителей экрана, включая флаг, звездное небо и квадриллион текстовых эффектов в стиле *Матрицы*. Если вам повезет, вы также получите несколько 3D-хранителей экрана, например, вращающийся текст и тоннель. Но действительно впечатляющие вещи зарыты поглубже. Попробуйте коллекцию *RSS-GLX* – Really Slick Screensavers (Реально Крутые Хранители Экрана). Это набор OpenGL-приложений, которые можно запускать самостоятельно или включить в *Xscreensaver* и превратить обычные хранители экрана демонстрашками времен Speccy [ZX Spectrum, – прим. пер.] размером в 1 КБ.

Установив RSS-GLX, вы обнаружите в каталоге **/usr/lib/xscreensaver** 19 новых двоичных файлов. Здорово, что ознакомиться с их эффектами можно, запустив их как обычные программы – в стандартном окне приложения. Каждый хранитель экрана имеет несколько опций для управления скоростью и сложностью. Любопытствуете, что умеет “Cyclone”? Введите

```
/usr/lib/xscreensaver/
cyclone --help
```

Коллекция, понятное дело, требует от видеокарты поддержки 3D-ускорения. Обязательно посмотрите Euphoria – завораживающе-прекрасный водоворот эффектов освещения и прозрачности, а также Skyrocket – вихревое панорамное пиротехническое шоу. Если хранитель экрана превратит вашу машину в черепаху, проверьте настройки согласно описанной выше процедуре и отключите часть эффектов.



Лишь немногие из имеющихся хранителей экрана. Посмотрите их в действии – обалдеете.

INCOLLECTOR



Информационный завал? Этот ловкий инструмент раскопает все

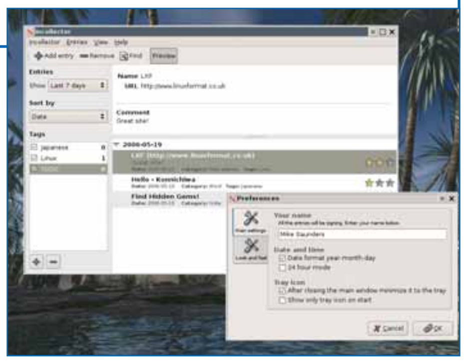
ИНФОРМАЦИЯ

ЧТО: Информационная коллекция
ГДЕ: www.incollector.devnull.pl
ЗАЧЕМ: Упорядочить вашу жизнь.

Бесчисленное множество Linux-приложений управляет коллекциями книг, фильмов, музыки и игр. Но каждое из них предназначено для объектов одного типа, и это осложняет интеграцию хранимой в них информации. Наша следующая жемчужинка не такова. *Incollector* позволяет не только сортировать информацию вроде ежедневных заметок и памяток. Это хорошо спроектированная и невероятно полезная программка, и хотя она создана под *GTK*, но будет прекрасно работать и в любом другом рабочем окружении или оконном менеджере.

Запустите *Incollector* и введите свое имя – в главном окне возникнет пустой лист. Щелкните на пункте Add Entry (Добавить запись) на панели инструментов, чтобы создать первую info-note (информационную запись). Появится выпадающий список, предлагающий выбрать категорию записи: web-адрес, перевод с другого языка, серийный номер или кусок исходного кода. Если информация, которую вы хотите сохранить, не подпала ни под одну готовую категорию, просто сохраните ее как Notes (Заметки). Каждую запись можно также снабдить комментарием и рейтингом.

По завершении ввода ваша запись появится в главном списке. Чтобы облегчить организацию записей, вы можете создавать метки (см. панель слева), а потом щелкать по пункту Find (Найти) на панели инструментов и осуществлять по ним поиск. Полезна возможность держать на экране записи только за последние несколько дней, это позволит избежать беспорядка от накопления десятка-другого записей.



Щелкайте на звездах для оценки информации по степени важности.

EASYTAG



Осточертели имена файлов вроде mus_0000025.mp3? Тогда прочитайте о...

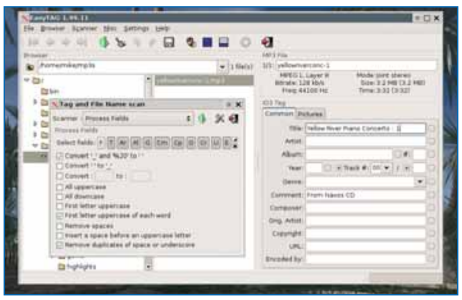
ИНФОРМАЦИЯ

ЧТО: Редактор тэгов музыкальных файлов
ГДЕ: http://easytag.sourceforge.net
ЗАЧЕМ: Создание музыкальных меток без усилий

Вы, вероятно, знаете, что MP3, OGG и файлы других музыкальных форматов содержат тэги – краткие сведения о песне (например, жанр и дата). Однако при конвертации музыки с CD эту информацию обычно надо вводить вручную – занятие нудное, и многие из нас им пренебрегают. Трижды поприветствуем *EasyTag*, сберегающий дни путем предельной автоматизации процесса тэгирования файлов – независимо от размера вашей фонотеки.

Например, *EasyTag* может обратиться к FreeDB – online-базе данных CD – для получения информации о песне и использовать ее для пометки файлов или заполнить поля, проанализировав каталог расположения и имена музыкальных файлов. *EasyTag* даже выполняет обратную операцию: если ваши файлы снабжены тэгами, но имена файлов бессмысленны (типа **mus156.mp3**), можно переименовать их *en masse*, взяв данные из тэгов (по типу **Автор-имя_песни.mp3**).

EasyTag поддерживает ID3 тэги MP2 и MP3 файлов, а также *Ogg Vorbis* и FLAC. Чтобы улучшить восприятие имен файлов, можете преобразовать буквы в верхний или нижний регистры; да еще можно проигрывать файлы прямо из приложения, любым проигрывателем.



***EasyTag* осуществляет полный контроль над тэгами.**

>>

<< TOMBOY



Записная книжка на стероидах!

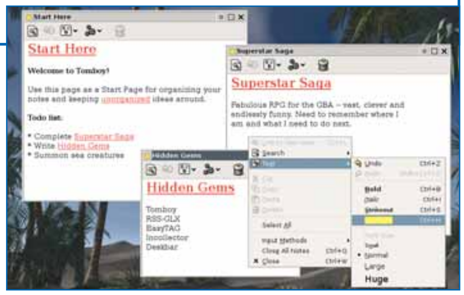
ИНФОРМАЦИЯ

ЧТО: Записная книжка в стиле wiki
ГДЕ: **www.beatniksoftware.com/tomboy**
ЗАЧЕМ: Создать свой собственный частный wiki

Представьте, что вы сидите за компьютером, и кто-то присылает вам список полезных Linux web-серверов. Вы перемещаете их в закладки, но хотите добавить к ним кое-какие заметки – например, что они содержат и как часто обновляются. Поэтому вы открываете *Leafpad* или *Emacs* и начинаете набирать заметки, пытаясь создать некое подобие структуры. Немного погодя это выходит из под контроля – тут-то и нужен *Tomboy*. Этот инструмент для заметок связывает вместе информацию и замечания подобно wiki, вместо сваливания всего в один большой файл.

Откройте *Tomboy*, и вы увидите небольшое окно для заметок, в котором вы можете немедля начать набирать. Сделайте пару заметок, но не старайтесь разместить все в одном месте – вы можете использовать связи (ссылки) для разбивки и систематизации информации. Например, вы можете набрать *В среду: сходить в магазин*, затем выделить слово *магазин*, щелкнуть на нем правой кнопкой мыши и выбрать Link To New Note (Ссылка на новую заметку). Теперь *магазин* имеет свою собственную страницу. Пример простой, но какого бы рода заметки вы ни делали, вы поймете, что такая система связей – прекрасный способ отсортировать информацию быстро и эффективно.

Tomboy также предоставляет функции простого форматирования (шрифт, кегль и так далее) – лишний довод против обычных текстовых редакторов. К изумлению, *Tomboy* все еще не включен во многие дистрибутивы – вот уж вправду секретное оружие Linux...



Tomboy **имеет средства HTML-экспорта для загрузки в сеть.**

ION



Используйте территорию экрана по максимуму

ИНФОРМАЦИЯ

ЧТО: Оконный менеджер
ГДЕ: **http://modeemi.cs.tut**
ЗАЧЕМ: Наконец-то – способ покончить с хаосом на рабочем столе

Мы просто обязаны были включить этот оконный менеджер: он действительно может изменить стиль работы. Не при помощи украшательства или функций и настроек, которые вам никогда не понадобятся, но применением принципиально иного подхода к рабочему столу. Большинство «нормальных» оконных менеджеров помещают приложения во всплывающих окнах, которые можно свободно перемещать по рабочему столу, а *Ion* использует мозаику окон – то есть окна на экране жестко закреплены и никогда не перекрываются.

Выглядит странно? Хорошо, представьте, что у вас открыт текстовый редактор вместе с Web-браузером и IRC клиентом. В традиционном оконном менеджере вы должны скакать с одного окна на другое или вручную разместить их на экране так, чтобы все они были видны одновременно. *Ion* делает это автоматически – что и означает термин «мозаика». Ваша мышь вздохнет свободнее, тем более что в *Ion* есть и чисто клавиатурный режим.

Ion настраивается при помощи языка скриптов Lua и снабжен превосходной документацией (введите **man Ion**). Рекомендуем всем пользователям Linux попробовать поработать в нем – возможно, он станет для вас идеальным рабочим окружением, особенно если вы программист.



Ion **демонстрирует функцию мозаики – перекрывающихся окон нет!**

DESKBAR



Поиск переходит на качественно новый уровень

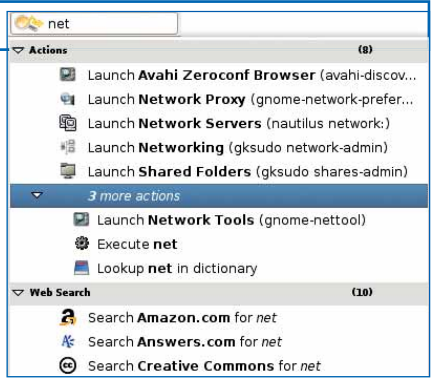
ИНФОРМАЦИЯ

ЧТО: Апплет строки поиска
ГДЕ: **http://raphael.slinckx.net/deskbar**
ЗАЧЕМ: Извлечь из поиска максимум.

До появления *Beagle* поиск файлов в Linux ограничивался сканированием списка имен файлов, не вникая в их содержиме. Теперь *Beagle* применяется довольно широко, но он открыл дверь и другим инструментам поиска для ПК, например, *Deskbar*. Это небольшой апплет, который сидит в панели Gnome, дожидаясь ввода; а дождавшись, выводит все хоть как-то с этим вводом связанное.

К примеру, наберите *тема* – он выведет не только файлы, содержащие это слово (еще пару лет назад вы другого и не ждали бы), но и связанные с «темой» программы – допустим, менеджер тем – наряду со ссылками, электронными письмами, IM-сообщениями и многим другим. Качество поиска зависит от установленного ПО: *Deskbar* использует встраиваемые модули (plugins), написанные на Python, чтобы привлечь дополнительные инструменты. Наилучшего результата можно добиться при установленном *Beagle*.

Deskbar также совершает поступки: наберите электронный адрес, и он откроет окно редактора писем; введите URL, и он запустит браузер. Таким образом, он хорош для любых «настольных» задач, доступных из одного графического элемента с текстовым вводом.



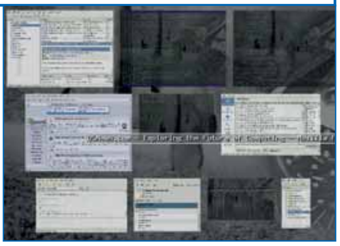
Deskbar **– приложение для Gnome 2.14. Вот что вы получите при вводе команды 'net'.**

SKIPPY

Панель задач – ну это же из 1990-х...

ИНФОРМАЦИЯ

ЧТО: Переключатель программ
ГДЕ: **http://thegraveyard.org/skippy.php**
ЗАЧЕМ: Получить Exposé для компьютера с Linux



Уменьшенные окна приложений в *Skippy* упрощают переключения между ними.

Если вам не доводилось видеть Exposé на Apple OS X, можете попробовать описываемое приложение – просто в порядке опыта работы в другом стиле. OS X – рабочий стол, предмет имитации для многих разработчиков, а Exposé – одна из его наиболее известных функций: это инструмент переключения программ, показывающий стартовавшие приложения в виде миниатюр. Можете не рыться в панели задач, а просто вызвать Exposé и щелкнуть на миниатюре необходимого приложения. В Linux есть несколько реализаций этого инструмента, но *Skippy* делает все наилучшим образом.

К счастью, для работы *Skippy* требуются только библиотеки *X11* и *Imlib2*. Запустите его из командной строки и нажмите клавишу F11. После небольшой паузы на создание миниатюр работающих программ вы увидите мозаичное изображение ваших окон, любое из которых можно выбрать мышью и щелкнуть на нем. *Skippy* уживается с большинством оконных менеджеров, хотя некоторые могут перехватывать у него клавишу F11, тогда ее придется переназначить.

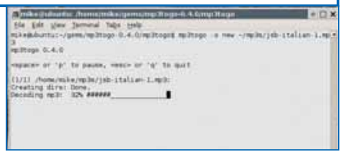
Skippy не столь подвижен, как Exposé в OS X; но зато не требует сложной процедуры установки или специального рабочего стола, то есть вы можете использовать его немедля! Мы были удивлены его малой известностью – сейчас нечто подобное умеет делать только сходная функция в Compiz

MP3TOGO

Выжмите все из вашего портативного плейера

ИНФОРМАЦИЯ

ЧТО: MP3-перекодировщик
ГДЕ: **http://puddle.ca/mp3togo**
ЗАЧЕМ: Больше песен в том же объеме.



Как консольное приложение, *Mp3togo* не столь приятен на вид, но он предоставляет полезную информацию, показывая прогресс.

Персональные MP3-плейеры ныне невероятно дешевы – 50-60 долларов за 256-МБ устройство. А если ваш плейер имеет меньшую память или вы просто хотите втиснуть в него больше песен? Перекодировка музыки при более низком битрейте или большей степенью сжатия означает большие хлопоты. Но ваш тайный друг *Mp3togo* автоматизирует этот процесс – можете заняться другими делами, а затем вернуться и запихать 25-минутную версию Blue Monday себе в плейер.

Установите *Mp3togo*, распаковав архив **Mp3togo_0.4.0.tar.gz**, зайдите в созданный каталог и от имени суперпользователя выполните команду **python setup.py install**. Теперь можно приступать к конвертации ваших файлов – причем сколь угодно большого числа, хотя если вы замахнетесь на сотни файлов, то потребуется немало времени (у нас при частоте 1 ГГц преобразование 6-МБ mp3-файла в 3 МБ заняло около 5 минут). Вот команда конвертации:

```
Mp3togo -o output-directory file1 file2 file3
```

Замените **output-directory** именем каталога, куда следует поместить перекодированные файлы, а **file1**, **file2** и т.д. – именами ваших mp3 (или используйте * для выбора всех файлов).

Программа преобразует файлы в WAV, нормализует их для лучшего качества звучания, перекодирует в Lame с большим сжатием и заполнит тэги. Voilà – ваши исходные музыкальные файлы почти уполовинились, лишь с небольшой потерей качества. И все готовы зазвучать!

LINKS

Web-серфинг без графического интерфейса? Свежая идея

ИНФОРМАЦИЯ

ЧТО: Web-браузер
ГДЕ: **http://links.sourceforge.net**
ЗАЧЕМ: Быстрый, элегантный и не требует X



Google News в *Links*. Длинные страницы прорисовываются в *Firefox* в четыре раза дольше, даже если отключить графику!

Не многие знают, что работать во всемирной паутине можно и без графического интерфейса. А еще меньше умеющих это делать. Однако разработчики *Links* и его ответвлений, держась в сторонке, релиз за релизом делают web-серфинг в текстовом режиме все удобнее. Но зачем же путешествовать в Интернет в текстовом режиме?

Важнее всего то, что это может спасти ситуацию, если вдруг отказал запуск X-ов. Если вы просто обновляете свой дистрибутив и вам понадобился новый драйвер, можете использовать *Links* в текстовом терминале и скачать его с web-сайта изготовителя видеокарты. Или, если при старте X возникла неизвестная ошибка, зайти на Google и найти решение проблемы. И потом, страницы отображаются молниеносно (благодаря игнорированию рисунков), а значит, это прекрасное средство просмотра Slashdot и других серверов новостей.

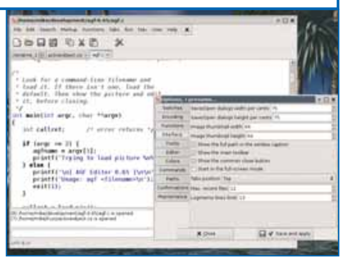
Links, а также его варианты E*Links* и *Links2*, имеют примитивную поддержку JavaScript и cookies – просто удивительно, сколь много сайтов можно просматривать без проблем, включая Gmail (HTML-режим) и даже сайт *LXF*. После запуска *Links* в терминале, нажмите F10 и попадете в главное меню, а там все просто: стрелки курсора для перемещения и Enter для выбора. Вряд ли вы будете использовать *Links* постоянно, но обнаружите, что это лучший способ просматривать любимые сайты.

TEA

Поразительное число функций для обработки текста

ИНФОРМАЦИЯ

ЧТО: Текстовый редактор
ГДЕ: **http://tea.linux.kiev.ua**
ЗАЧЕМ: НУИНУ текстовых редакторов

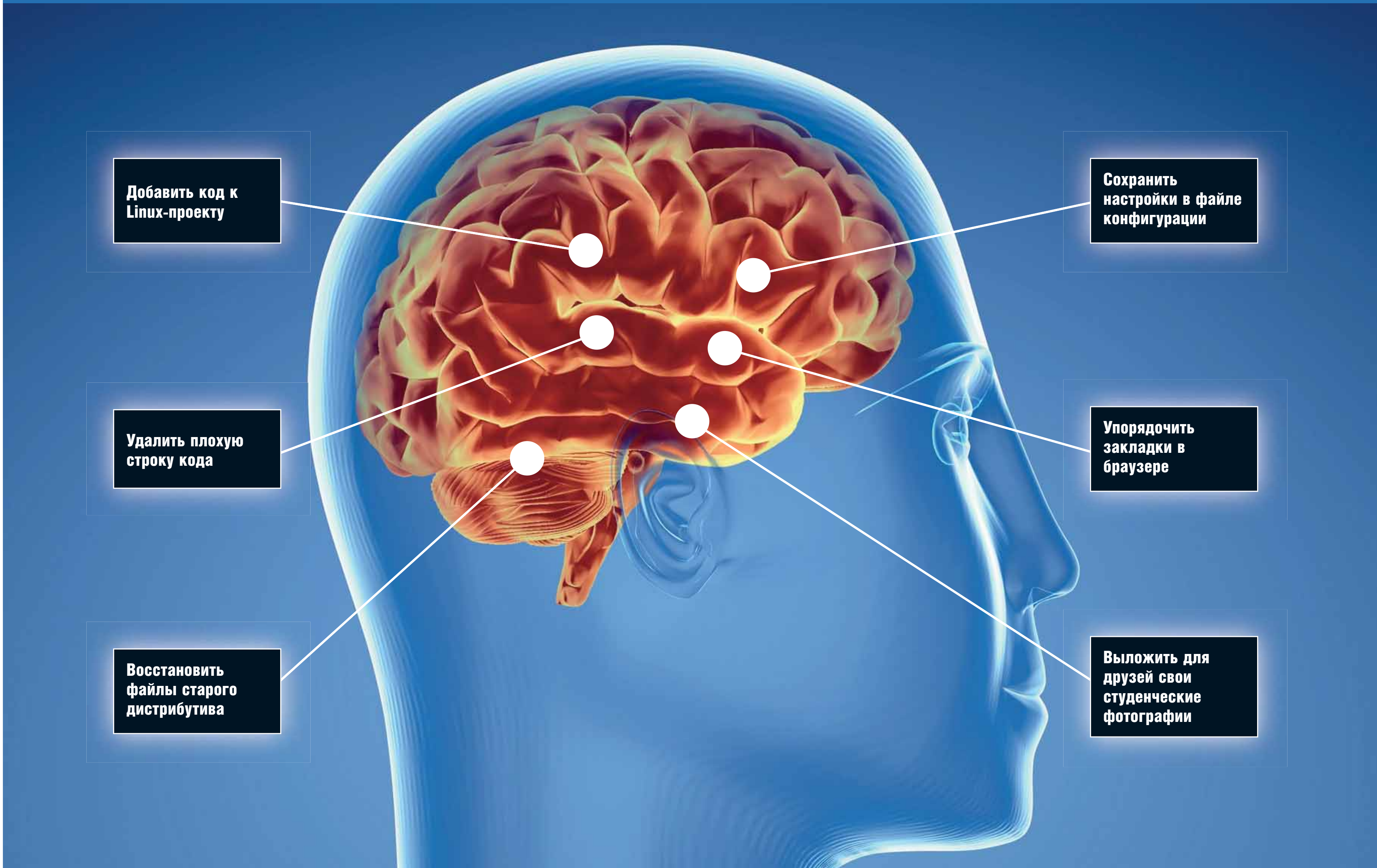


Если вас тянет на подвиги, откройте окно *Prefs* для продвинутой настройки.

Мы рады, что осветили факелом *LXF* это небольшое приложение. Если вы давний пользователь Linux, то, вероятно, слышали о *Emacs*, *Vim*, *Gedit*, *Kate* и прочих популярных текстовых редакторах. Но бьемся об заклад, что вы, как и мы до недавнего времени, никогда не слыхивали о Tea. Этот неброский редактор с Украины вместил огромное количество функций в обычном бинарном пакете размером менее 500 КБ [Особенно приятно отметить, что разработчик TEA Петр Семилетов является одним из постоянных авторов нашего журнала; читайте его обзор аудиоплейеров на стр.22, – прим. ред.].

Tea выглядит и ведет себя как обычный текстовый редактор *GTK*, но поройтесь в меню – и вы обнаружите штабеля вкусностей. Среди них: календарь, менеджер файлов, средства IDE, проверка орфографии, подсветка синтаксиса, инструменты HTML-конвертации, импорт OpenDocument, встроенный просмотрщик изображений, и т.д., и т.п. Вы даже найдете (и мы не шутим!) функцию преобразования сигналов азбуки Морзе.

Во многих других приложениях все это выглядело бы как свалка функций, что верно и здесь, и не все они будут полезны каждому. Но разработчики сумели упаковать эти функции в значительно меньший объем по сравнению с другими функциональными редакторами, а это меняет угол зрения. Tea заслуживает большего внимания, чем ему уделяется, и если вас тяготит размер Emacs и Vim, это первый кандидат на смену. LXF



Жизнь надо прожить в SUBVERSION

Однажды дождливой ночью **Грэм Моррисон** нечаянно скинул на сервер *Subversion* свой домашний каталог – и вдруг осознал всю полезность этой идеи. Вам тоже стоит так сделать.

НА ДИСКЕ

- Subversion 1.3.1
- KDESvn 0.8.4

А зачем? Зачем вообще использовать инструмент разработчиков для хранения собственных файлов? Вы, наверное, знаете, что *Subversion* обеспечивает доставку актуальной копии исходного кода совместно работающим над ним через сеть программистам. Но лежащая под этим технология также полезна и для других целей. *Subversion* – сервер, хранящий в репозитарии изменения, произведенные над группой файлов. Он позволяет и удалить нежелательное программное обеспечение, и установить его заново. Это не сложнее, чем набрать команду.

Для домашнего каталога это означает, что всегда имеется его интеллектуальная резервная копия. Можно не только спасти случайно удаленные данные, но и выбрать возможность восстановления до предыдущего состояния; или удалить неудачный вариант конфигурации; или посмотреть, как выглядели закладки в браузере год назад. Если вы работаете на нескольких машинах, то ваш домашний каталог синхронизируется с каждой из них – чем не изящное применение *Subversion*? Любые сделанные изменения тут же попадают на все машины, гарантируя однозначность копии **/home**. Забавно также выкатить часть вашего репозитария *Subversion* на публику, в форме Linux-блоггинга.

Так что «субвертите» ваш домашний каталог на здоровье, для этого не надо быть суперхакером. Все, кто не боятся командной строки, могут попробовать, а мы вам поможем.

Установим сервер Subversion

Говоря попросту, *Subversion* – подобие умной базы данных, используемой программистами для отслеживания изменений в проекте на протяжении всего его существования. Без *Subversion* группе разработчиков одного проекта приходилось бы посылать все изменения лидеру проекта, а он затем вносил бы их в проект. *Subversion* заменяет лидера проекта сервером, и изменения вносятся автоматически.

Важно, что сервер *Subversion* хранит только отличия между версиями, а каждое подтвержденное изменение называется пересмотром (revision). Проходя по пересмотрам вперед и назад, можно раздобыть копию любой стадии проекта.

Для *Subversion* безразлично, если вместо исходного кода вы загрузите свой домашний каталог; его дело – хранить изменения. Однако имеется несколько загвоздок. Прежде всего, проблемы вызываются большими двоичными файлами. Не то чтобы *Subversion* их не поддерживал, но они реально замедляют систему, так как *Subversion* вынужден их сравнивать, отслеживая разницу. Возможно, стоит держать большие файлы вне репозитария.

Руководство по установке

Труднее всего установить *Subverion* и заставить его работать в первый раз. Сервер не самая простая вещь, но ничего невозможного тут нет. Если ваш сервер располагается на той же машине, что и домашний каталог, то вряд ли понадобится настройка. Зато желание получать доступ к своему репозитарию через сеть намного все усложнит.

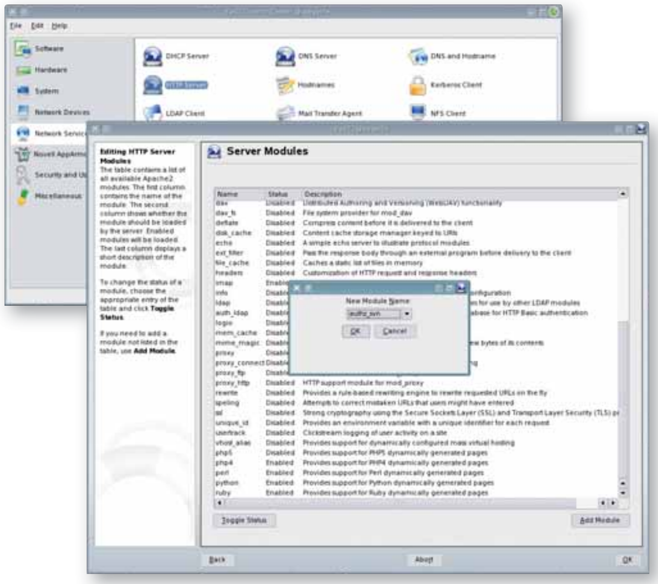
О том, как установить собственный сервер, говорилось в LXF70, но вкратце опишем процесс, чтобы вы смогли начать. Каждый солидный дистрибутив включает пакет *Subversion*, который легко устанавливается с помощью менеджера пакетов. Войдите в менеджер пакетов вашего дистрибутива, поищите пакет **Subversion-server** и установите его. Если вы уже использовали *Subversion*, можете установить дополнительные модули *Apache* и получить доступ к репозитарию через HTTP – ищите пакеты **dav_svn** и **authz_svn**.

Пользователей SUSE ждет проблема: сервер не входит в поставку SUSE 10.0. Поэтому поищите пакет **Subversion-server-1.2.3-2.i586.rpm** в интернете (или на диске к журналу) и установите его от имени суперпользователя с помощью команды **rpm -Uvh *.rpm**. Если вы используете *Apache*, можете также сконфигурировать его через *Yast*, выбрав Network Srvices > HTTP Server и нажав на Modules. Add Module позволит вам установить оба 'svn' модуля и модуль WebDav.

Перезапустите *Apache*, набрав от лица root в командной строке **/etc/init.d/Apache2 restart**. SUSE опять не повезло: возникает ошибка

```
undefined symbol: dav_svn_split_uri
```

Все потому, что модули *Apache* должны загружаться в правильной последовательности. Откройте **/etc/sysconfig/Apache2** в любимом текстовом редакторе, найдите строку, начинающуюся с 'APACHE_MODULES', и позаботьтесь о следующем порядке загрузки модулей: **dav dav_svn authz_svn**. Теперь *Apache* перезапустится без проблем.



В SUSE возможен HTTP-доступ к репозитарию *Subversion*, если добавить модули при помощи Yast.

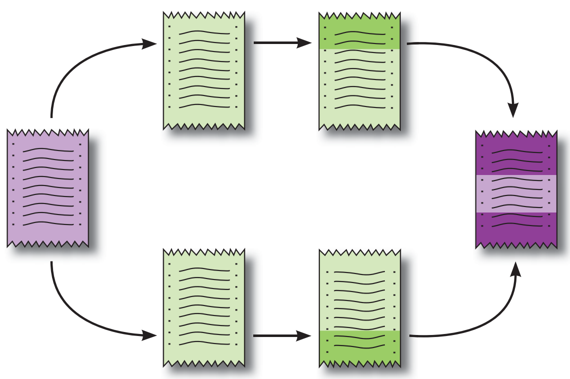
Выбор протокола

Доступ к репозитарию *Subversion* осуществляется с помощью URL. Существует три протокола доступа. Самый простой для конфигурации – URL, начинающийся с 'file:///'. По нему доступ осуществляется к репозитарию, размещенному на вашей файловой системе – просто проверьте, что слэшей три. Для такого доступа настройки вовсе не требуется. Установите *Subvesion*, и за работу!

Второй способ также легко реализовать, и он заключается в запуске отдельного TCP сервиса, для обслуживания запросов к серверу *Subversion*. Сложным это кажется только на первый взгляд. Проверьте, что установлена утилита *Svnserve*, создайте пользователя и группу 'svn' командами *useradd* и *groupadd*; затем запустите сервис *Svnserve*, набрав **/etc/init.d/svnserve start**. Под Mandriva и Red Hat он запускается командой **service svnserve start**.

Третий способ – использование HTTP через сервер *Apache*. Для ясности, мы в своих примерах будем использовать file:///, а вы можете избрать способ доступа по своему вкусу.

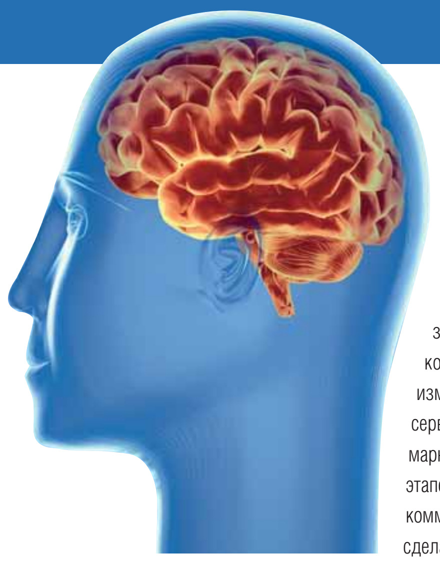
>>



1 2 3 4

Разработчики, использующие *Subversion*, могут работать над одним проектом одновременно. [1] Старая версия файла находится в репозитарии. [2] Два разработчика скачивают один файл. [3] Оба работают параллельно над своими частями файла, добавляя в них свой код. [4] Когда они отошлют результаты обратно на сервер, произведенные изменения сольются в один файл.

/home переезжает



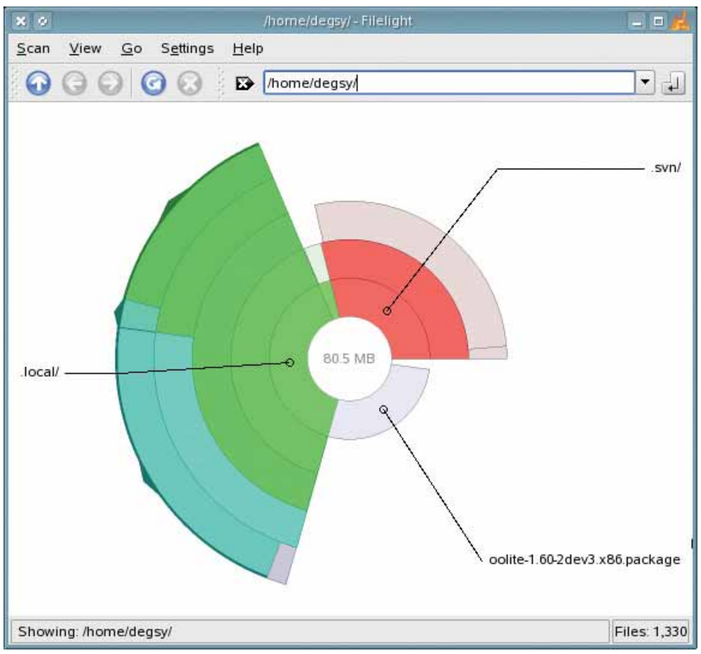
После запуска сервера следующей задачей является заполнение репозитария данными из вашего домашнего каталога. Для начала создайте репозитарий с помощью команды *svnadmin*, определив, где именно хранить данные. (Заметим, что в последующих примерах мы проводим весь процесс для пользователя degsy.)

```
svnadmin create /srv/degsy
```

Команда *svnadmin* не использует префикс протокола (file:///), потому что всегда запускается на стороне сервера.

Теперь надо скачать репозитарий с сервера, то есть просто создать локальную копию данных, размещенных на сервере. На текущий момент на нашем новом сервере никаких данных нет, но процедура скачивания создаст в вашем домашнем каталоге временное хранилище для файлов конфигурации, которые *Subversion* будет использовать для отслеживания изменений в ваших данных.

Приступим. Зайдите на вашу систему как суперпользователь и перейдите в каталог **/home**. Далее переименуйте каталог того пользователя, который собирается использовать *Subversion*. В нашем случае **home/degsy** станет **home/foo**. Изменение имени необходимо, чтобы не произошло перезаписи данных, потому что, когда мы выкачиваем домашний каталог из *Subversion*, он замещает исходный каталог.



Помните, что *Subversion* копирует все содержимое репозитария в скрытый каталог .svn, это удваивает требуемый объем дискового пространства.

Опасаться нечего: в конце концов, мы всегда сможем вернуть наши данные из **/foo**. Вам надо выполнять операции от лица суперпользователя, потому как только он имеет права на проведение изменений в каталоге **/home**.

Переименовав каталог, создайте копию данных сервера *Subversion*, а заодно и файлы конфигурации:

```
svn checkout file:///srv/degsy
Checked out revision 0.
```

Каждый раз, когда вы производите изменение в файлах, находящихся на сервере *Subversion*, номер ревизии (пересмотра) увеличивается на единицу. *Subversion* использует атомарные изменения: неважно, сколько файлов успело поменяться в вашем домашнем каталоге – когда репозитарий обновится, все изменения пройдут как один пересмотр. В показанном выше примере число пересмотров равно нулю, потому что мы еще ничего не делали с репозитарием. В **/home** вы увидите каталог с именем вашего пользователя (то есть **/home/degsy**), и если вы поинтересуетесь содержимым этого каталога, то обнаружите, что в нем содержится всего одна запись. Чтобы просмотреть ее, наберите

```
ls -al degsy
```

и увидите следующее:

```
drwxr-xr-x 7 root root 4096 2006-05-11
20:30 .svn
```

Это скрытый каталог *Subversion*, где будут храниться все изменения, а также исходные версии всех файлов. Теперь надо скопировать этот каталог **.svn** в исходный домашний каталог – который теперь называется **/home/foo**; именно он будет наполнять репозитарий *Subversion*. Далее, удалите скачанный с сервера каталог и верните исходному домашнему каталогу его предыдущее имя:

```
cp -rf degsy/.svn foo/
rm -rf degsy
mv foo degsy
chown -R degsy:users degsy
```

Теперь все на своих местах, и можно предпринять важный шаг загрузки домашнего каталога *Subversion* на сервер. Начнем с команды *add* – она добавит файлы, которые *Subversion* посчитает новыми (то есть все). После add используем команду commit, которая проверит все изменения и загрузит их на сервер как неделимую атомарную ревизию. На этом этапе вас попросят написать комментарий, описывающий сделанные вами изменения. Программисты используют комментарии для описания сделанных изменений. Комментарии пригодятся и для вашего домашнего каталога, особенно если перемены значительны: например, сменился дистрибутив или оконный менеджер. Если вам необходимо вернуться к конфигурации до обновления, просто просмотрите комментарий.

Наберите **svn add ***, чтобы добавить все файлы вашего домашнего каталога на сервер *Subversion*:

```
svn add *
A bin
A Documents
A Documents/.directory
A images
A (bin) images/IMG_0192.JPG
```

Двоичные файлы

Subversion теперь переберет все файлы, добавляя каждый из них в список файлов, которые необходимо закачать на сервер. Двоичные и текстовые файлы обрабатываются различным способом. Вот почему изображение в выше показанном примере помечено как **bin**. Изменения в двоичные файлы вносятся иначе, чем в текстовые, когда *Subversion* просто хранит прямые изменения между одной версией и следующей. Символ **A** в начале каждой строки означает, что каждый файл будет добавлен добавлен в репозитарий, но этого не произойдет, пока вы не запустите команду *commit*, решившись принять изменения окончательно. Также вас попросят написать комментарий, описывающий смысл изменений.

Получится следующее:

```
svn commit
Adding Documents
Adding Documents/.directory
Adding bin
Adding images
Adding (bin) images/IMG_0192.JPG
Transmitting file data.
Committed revision 1.
```

Время выполнения этого процесса зависит от количества произведенных изменений и от скорости соединения с сервером

Subversion. Когда вы в первый раз подтверждаете изменения вашего домашнего каталога, то наверняка потребуется значительное время для копирования всех ваших данных.

Вы, возможно, заметили, что есть одна жизненно важная группа файлов, которая не обновилась – а именно, скрытые файлы (их имена начинаются с точки «.») вашего домашнего каталога. Почему? Потому что это вызовет конфликт с собственным скрытым каталогом *Subversion* (**.svn**). Если вы знаете, какие каталоги вам нужны, добавьте их вручную. Следующая команда добавит, к примеру, конфигурационные файлы Gnome:

```
svn add .gnome2
A .gnome2
A .gnome2/accels
svn commmit
Committed revision 2
```

Альтернатива – перемещение скрытых каталогов (исключая **.svn**) в отдельный каталог и использовании простого скрипта для воссоздания символьных ссылок. Небольшой недостаток использования *Subversion* состоит в том, что, когда вы добавляете и удаляете файлы и каталоги, требуется повторять процесс для репозитария *Subversion* с помощью команд *svn add* для добавления и *svn del* для удаления, а затем выполнять *commit* для подтверждения изменений. Но трудно ли заменить эти команды скриптом?

10 ШАГОВ ПО ЖИЗНИ С SUBVERSION

Проект настолько разумно организован, что у новичков просто нет шансов запутаться. Вот предпринятые нами действия, по шагам:

- **Установите *Subversion* и выберите протокол доступа.**
- **Создайте репозитарий *Subversion* для вашего домашнего каталога, например, /srv/user**
- **Переименуйте домашний каталог пользователя /home/user в /home/foo**
- **Сделайте локальную копию нового репозитария *Subversion* /srv/user**
- **Проверьте, что в /home создался новый каталог user, включая скрытый каталог .svn**
- **Скопируйте .svn в /home/foo**
- **Удалите вновь созданный каталог /home/user**
- **Переименуйте /home/foo в исходный /home/user**
- **Загрузите /home/user в репозитарий *Subversion***
- **Вручную добавьте скрытые файлы**

Приемы работы

Итак, вы используете *Subversion* для хранения домашнего каталога. Посмотрим, что полезное можно сделать благодаря этому.

1 Защитить данные

Если вам надо получить доступ к домашнему каталогу через Интернет, то использование протокола **'svn://'** может подвергнуть вас риску атаки прослушивания, когда кто-то перехватывает ваши данные во время их передачи. Решение состоит в использовании безопасной оболочки SSH. Просто замените 'svn' в URL на 'svn+ssh'. Вам придется использовать учетную запись пользователя, существующую на сервере, а также проверить, что этот пользователь имеет право на доступ к репозитарию. Чтобы определить пользователя в URL, наберите:

```
svn list svn+ssh://degsy@hostname/srv/degsy.
```

2 Везде как дома

Поместив свой домашний каталог на сервер *Subversion*, вы можете иметь те же файлы и настройки на нескольких машинах хоть на другом краю Земли. Так как машины синхронизируются с сервером, то вы можете быть уверенными, что используете актуальные файлы. Когда вы находитесь в каталоге **/home** на новой машине и хотите создать копию репозитария, то проверьте имя сервера и наберите

```
svn checkout svn+ssh://hostname/srv/degsy
```

Эта команда создаст копию домашнего каталога degsy в текущем местоположении. Любые изменения, сделанные вами на этой машине, надо отослать на сервер и подтвердить. По команде **svn add *** система проигнорирует старые файлы и добавит только новые.

3 Плановые правки

Обновление версии вручную – это фирменный рецепт, как лишиться файлов или синхронизации каталогов. Как ни печально, простого способа синхронизировать новые или удаленные папки и файлы в автоматическом режиме не существует – требуется все время набирать *svn add* и *svn del*; но самую длинную команду svn commit можно автоматизировать.

Добавьте команду *commit* в системную таблицу cron с помощью команды crontab. Предварительно создайте таблицу для пользователя с помощью команды **crontab -e**, а затем можно добавить запись для запуска команды *commit*:

```
15 00 * * * svn commit -m
  "Automated commit"
```

Команда commit запустится в 00:15, а параметр -m говорит, что мы не хотим взаимодействовать с процессом для добавления комментария к изменениям.

4 Спасение рядового файла

Еще одна привлекательная черта такого репозитария – восстановление случайно удаленных или потерянных данных. Если вам требуется старая версия данных, то должна быть и возможность просмотреть изменения. Этой цели служит команда *diff*, которая точно скажет, что и где изменилось в файле.

Например, следующая команда в точности отобразит изменения между 5 и 6 ревизиями файла **bookmarks.html**:

```
svn diff -r 6:5 file:///srv/degsy/bookmarks.html
```

Существует несколько способов восстановить файл с помощью этих данных, и самый безопасный из них – сохранить старую версию файла в новый каталог внутри домашнего каталога. С помощью команды *svn copy* укажите номер ревизии и путь, куда копировать:

```
svn copy -r 6 file:///srv/degsy/bookmarks.html new.html
```

5 Резервное копирование

Грош цена вашему репозитарию, если данные вовремя не резервируются, особенно если ваш домашний каталог всегда временный. Отличное решение – запуск на сервере команды **svnadmin dump**: она выводит все содержимое репозитария на экран, но ведь вывод можно перенаправить и в файл!

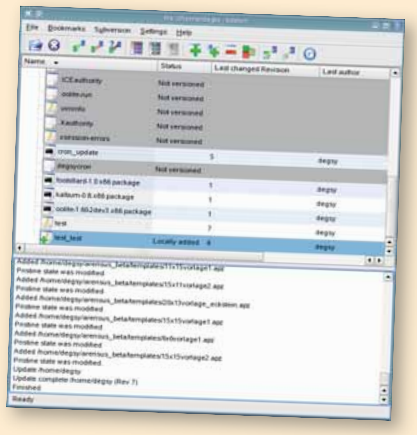
```
svnadmin dump >svn_backup
```

Проверьте, что svn_backup хранится в надежном месте, и не забывайте регулярно проводить эту процедуру. Восстановление данных из резервной копии обеспечит следующий трюк:

```
svnadmin load /srv/degsy < svn_backup
```

Что дальше?

На радость хакерам, существует много способов для написания собственных решений в виде скриптов – например, по части хранения скрытых файлов и создания резервных копий. *Subversion* был задуман не для вашего домашнего каталога, но он невероятно гибок – так чего ж теряться? LXF



KDESvn **избавит вас от консоли при работе с изменениями в репозитарии.**



LXF ИНТЕРВЬЮ

АЛЕКСЕЙ ГУРЕШОВ:
ЧЕЛОВЕК ИЩУЩИЙ

Генеральный директор проекта webalta о своей поисковой машине

Визитка LXF

Алексей Гурешов

Генеральный директор многообещающего проекта Webalta – нового игрока на рынке интернет-поиска, собирающегося навести шороху в рядах признанных фаворитов

ВОЗРАСТ	24
НАЦИОНАЛЬНОСТЬ	+
СТАЖ РАБОТЫ В LINUX	8
ЯЗЫКИ ПРОГРАММИРОВАНИЯ	читатель
КОЛИЧЕСТВО ПК	2
ДНЕВНАЯ НОРМА КОФЕ	чай
ПАР САНДАЛИЙ	1

ДЕВИС: «Использовать все шансы.»

LinuxFormat: Рынок поисковых сервисов достаточно насыщен. В мире широко известны MSN, Google. В России – Яндекс, Rambler, Апорт, наконец. Как у вас родилась идея «втиснуться» в эту тесную компанию? Рассчитываете ли вы получить какую-то определенную долю рынка или занять свою уникальную нишу?

Алексей Гурешов: Попробую объяснить. В мире существует множество почтовых сервисов, Yahoo, Mail..ru и т.д. Но почему в таком случае поисковиков должно быть всего два или три? Кто говорит, что этот рынок насыщен? Об этом говорят сами поисковики, которые просто гребут деньги, и им выгодно так утверждать. Послушайте, какие комментарии дают Яндекс, Rambler, Google и другие о насыщенности рынка. Понятно, что никто из них не хочет пускать на этот рынок других конкурентов.

Но что касается качества поиска, то очень многие недовольны текущими поисковиками. Все заполонили дорвеи, и поисковый спам является достаточно серьезной проблемой. И почему же в таком случае не попробовать свои силы и не создать достойного конкурента? Тем более, что уже есть серьезные наработки, которые предоставили нам финансовые возможности для создания поисковой машины.

Я слышал такую идею, что в ближайшие годы сможет выжить и укрепить свои позиции тот, кто сделает не просто поиск по сайтам, так как этого уже недостаточно, а тот, кто сможет предоставить результат в комплексе. В частности, вы вводите запрос «Владимир Путин», и вам сразу же демонстрируется его портрет, показывается его биография и т.д. Мы как раз планируем развиваться в эту сторону, и ведем активную работу в этом направлении.

Второй момент – борьба с поисковым спамом. Проблема эта угрожающая, и если вы посмотрите статистику по тулбару LiveInternet, которые считает переходы по сайтам, программы-дорвеи имеют всего лишь вдвое меньше трафика, чем весь «Яндекс». Возьмем в качестве примера всего два сайта – Блокнотик.ру и Поиск.ру. Один имеет 23 процента, а другой 17 процентов, при этом 100 процентов имеет сам «Яндекс». Надеюсь, вы можете себе представить масштабы бедствия?! Просто об этом мало кто говорит, так как все это можно считать негативом, неким черным PR для поисковых машин.

LXF: Сколько человек работает в вашей компании?

АГ: В данный момент у нас трудится порядка 70 человек. И мы продолжаем искать талантливых людей. Например, поиск программистов, особенно толковых – очень большая проблема. Планируем также увеличить операционный отдел с выводом в свет контекстной рекламы. В мыслях развивать отдел PR и брендинга.

LXF: Сколько финансов было вложено в развитие поисковой машины Webalta, и сколько еще будет инвестировано?

АГ: У нашей компании есть уставной капитал в размере 20 миллионов рублей. На эти деньги сейчас выпущены акции. Наша компания – это открытое акционерное общество. На самом деле, вложений было гораздо больше, так как мы брали займы у частных лиц, учредителей. Разумеется, эти средства планируется вернуть. В реальности сегодня только на основные средства потрачено порядка 10 800 000 рублей и до достижения самоокупаемости нам придется еще достаточно вложить. У нас запланированы довольно большие затраты, например, на продвижение. Мы ведь планируем занимать долю на рынке.

LXF: Хотелось бы перейти к техническим вопросам. Как Webalta устроена изнутри? Как я слышал, в ее основе лежат разработки на основе Open Source?

АГ: Да, в основе поисковой машины лежит Scientific Linux 4.2. Мы выбрали его потому, что собран этот дистрибутив на базе Red Hat Linux. Если последний платный и стоит немалых денег, то Scientific Linux распространяется совершенно свободно. Так как у нас довольно много серверов, то приобретать для каждого лицензию на Red Hat Linux было бы довольно накладно.

LXF: Но ведь у Red Hat есть ряд преимуществ, например – сертификация Oracle.

АГ: Нам это не нужно, так как мы используем только Linux, и у нас нет ничего чужого, даже база данных у нас своя собственная. Еще одна причина, по которой мы выбрали Scientific Linux состоит в том, что компания Red Hat самостоятельно обновляет ядро, пишет модификации, накладывает патчи, а в данном случае мы сами берем ядро с Kernel..org и оно отлично работает. Возникала лишь проблема с драйверами, так как некоторые производители, в частности Adaptec, предоставляют драйверы в бинарном виде, собранном исключительно под Red Hat.. Но впоследствии мы эту проблему полностью решили, и сейчас у нас все отлично работает. Поначалу мы хотели выбрать дистрибутив Fedora Core, но умные люди нас отговорили от его использования, так как на нем Red Hat только тестирует и обкатывает все нововведения.

LXF: Не было ли идеи использовать Debian, который традиционно считается наиболее стабильным и консервативным решением?

АГ: Да, мы смотрели много дистрибутивов. Нам еще очень понравился Gentoo.. Но уж так исторически сложилось, что человек, который собирал нам весь кластер, нас убедил в том, что надо использовать именно Scientific Linux. Этот человек работает в довольно известном московском учебном заведении и занимает там серьезную должность, так что мы приняли его слова на веру.

LXF: Вы упомянули о кластере. Интересно было бы узнать, сколько в нем узлов, и какой объем трафика проходит через него.

АГ: Узлов, то есть серверов сейчас около семидесяти. Трафик на самом деле очень большой. Все узлы соединены через гигабитный Ethernet. Вначале мы хотели сделать все на основе Intel, но потом у нас возникли проблемы. Так, нам потребовалось установить сервер с 16 гигабайт ОЗУ. Для интеловских серверов требуется специальная память, которая стоит очень дорого. Получалось так, что эти 16 гигабайт стоили столько, сколько два аналогичных сервера с 16 гигабайтами на базе AMD Opteron. Так что сегодня мы практически полностью перешли на Opteron. Как нам кажется, по соотношению цена/качество сегодня на рынке ничего лучше нет.

Тех серверов, которые у нас есть сейчас, нам хватит, по крайней мере, до конца года. Но уже осенью мы собираемся индексировать Европу, поэтому расширение технической базы неизбежно. Так что уже осенью серверная база вырастет втрое.

LXF: А кто поставляет вам серверы?

АГ: Мы сотрудничаем с двумя компаниями – Trinity Solutions и «Т-Платформы». По качеству и по уровню сервиса эти компании примерно равны.

LXF: На каком языке написан основной компонент поисковой машины – «паук»?

АГ: Вообще, у нас не только «паук», но вся система написана на C++. Реально у нас просто установлен дистрибутив Linux, поверх которого стоит собственная программа. Никакого Apache у нас также нет, мы используем собственный http-сервер. Вначале мы пробовали использовать MPI для коммуникации между серверами, но как выяснилось, для наших целей он совершенно не годился, так как постоянно грузил процессор, туда-сюда гонял нулевые биты.

LXF: Насколько я знаю, вы отказались от использования файловых систем, и у вас осуществляется запись прямо на «голый» раздел?

АГ: Непосредственно в самих инвертированных списках так предполагалось первоначально. Это очень важно для скорости и оптимизации, так как мы можем раскладывать данные на диск так, как это нужно нам, а не файловой системе. У нас возникали проблемы с сервером, который содержит оригинальные документы, первоначально мы планировали складывать их на сервере с файловой системой ReiserFS, однако подобная схема сразу же не сработала. Дело в том, что ReiserFS попросту не в состоянии записывать на диск 4000 мелких файлов в секунду. В результате нами создана собственная файловая система, специально ориентированная на подобные вещи. Сейчас это хранилище у нас занимает примерно 12 терабайт, и его легко можно расширить до 96 терабайт. Попросту возможности ReiserFS нам оказались не нужны.

LXF: А вы не думали о реализации собственной файловой системы в ядре?

АГ: Хороший вопрос. Сейчас мы Интернет делим по кускам, доменам. Это все работает до определенной стадии, когда у тебя 20-30 узлов. Но когда их число доходит до сотни, возникают проблемы. Дальше Интернет будет нужно делить уже по словам. Списки определенных слов хранятся на определенных серверах. Чтобы данная система устойчиво работала, нам как раз и потребуется собственная файловая система. Сейчас мы ищем специалистов, способных написать подобные вещи. Скорее всего, мы будем делать файловую систему, которая сама бы нам раскладывала информацию, да и не просто раскладывала, но и могла делать равномерное дублирование по серверам. Это даст нам возможность использовать много маленьких дешевых машин, когда у нас пойдет реальная нагрузка.

В принципе, та система, которая у нас есть сейчас, выдерживает 20 миллионов запросов. Если поставить вторую копию, это даст 40 миллионов. Для России этого хватит, но чтобы поддерживать 14–16 миллиардов, нужно делить все по-другому. Естественно, мы думали об этом с самого начала, и перевести систему на новые рельсы не составляет никакого труда.

LXF: У каждой поисковой системы свой собственный алгоритм. Кто разрабатывал ваш алгоритм?

АГ: Алгоритмы поиска, как таковые описаны в учебниках. Эта информация существует уже давно и ничего принципиально нового здесь не придумано. Формула, которую мы используем для расчета близости, текста ссылок и т.д. – это наша собственная разработка. Главная задача поисковых алгоритмов – тестирование поисковой машины. Ведь Интернет сам по себе далеко не идеальный, там полно накрутчиков и множество сайтов, которые неправильно устроены, не соблюдают стандарты и т.д. Одним словом, это большая куча мусора, которую нужно как-то структурировать. Люди пишут в Title и на страницы все, что хотят, а задача поисковика выбрать то, что нужно. Сейчас у нас есть свыше двухсот настроек. Каждый коэффициент на что-то влияет.

Так, мы умеем определять содержание страницы и т.д. Каждый коэффициент при изменении улучшает или ухудшает позицию страницы. Основная задача всех этих алгоритмов сводится именно к тестированию. Ведь как раньше тестировали поисковики? В них загружали некую идеальную базу документов, после чего добавляли эталонную базу вопросов и ответов, и после этого выравнивали. Нам кажется, что на текущий момент это уже устаревшая модель и тестировать надо непосредственно на Интернете. Мы уже создали эталон- >>



<< ную базу вопросов и ответов, когда например, по слову «погода» в первой двадцатке должны оказаться одни результаты, на фразу «официальный дилер Mercedes» другие и т.д. Подобных запросов сейчас уже собрано около тысячи. У нас есть средство автоматической настройки, которое восстанавливает все коэффициенты с целью получения наиболее точных результатов. Но на основной базе мы его пока не пробовали.

На днях мы перешли на новое ядро, исправив тем самым множество критических ошибок. Нам пришлось полностью обнулить базу, и теперь мы собираем ее заново. В последующие дни мы уже будем заниматься качеством.

LXF: Какой принцип ранжирования страниц вы считаете наиболее оптимальным?

АГ: Один из ключевых моментов при ранжировании – это ссылки. Все поисковики используют определенные условия и определенные функции для учета этих ссылок. «Яндекс» считает ссылки по доменам, Google по страницам, а у нас реализована собственная модель, в результате которой получилось нечто среднее. Кроме того, мы добавили целый ряд своих нововведений, в частности мы учитываем не только ссылки на доменах, но и дату регистрации этих доменов. Ведь даже если появился какой-то сайт, но он появился только вчера и на него уже наставили кучу ссылок, вероятность того, что он будет в первых результатах поиска довольно мала. Мы это называем «уровнем доверия» к сайту. Помимо этого мы смотрим и на другие параметры – WHOIS, DNS и т.д. Все это в той или иной мере необходимо для борьбы с поисковым спамом. Все-таки хочется эту борьбу как-то автоматизировать, потому что сейчас в Рунете она как-то не слишком автоматизирована, по той причине, что поисковики борются с этим явлением уже по принципу свершившегося факта. То есть, поисковые спамеры сделали что-то, их обнаружили и забанили. Конечно, пытаются создавать специальных роботов, которые понимают Java Script и будут ходить и детектировать все эти дорвеи, но это тупиковый вариант. Ведь тот же Java Script можно закодировать так, что ни один робот его не сможет распознать. Кстати, мы перешли на новое ядро и сбросили базу, теперь наращиваем ее. Далее займемся ее качественной отстройкой. Хотелось бы довести качество поиска до нормального состояния, чтобы его результаты были хотя бы сопоставимы с «Яндексом«.

Мы продолжаем разрабатывать новые интересное направление, например, собственный тулбар. Фактически он уже существует, и мы сейчас занимаемся шлифовкой мелких деталей.

LXF: Тулбар будет для всех браузеров?

АГ: Нет, пока только для Internet Explorer. Просто не все сразу.

LXF: Сегодня ваши поисковые базы заполнены не до краев. Когда планируется завершить наполнение?

АГ: Планируется, что к началу осени мы будем иметь одну из самых больших баз по российским документам. Текущая скорость индексации у нас составляет примерно сто миллионов уникальных страниц в сутки. Построить индекс по миллиарду страниц мы можем за десять дней. За август мы как раз планируем довести индекс до миллиарда. Вообще, русских страниц не так уж много, и довольно сложно бывает найти нужное количество страниц, с тем, чтобы построить самую большую базу.

LXF: Компания Google часть своего кода распространяет по открытой лицензии. Планируется ли нечто подобное в Webalta?

АГ: Пока не планируем. Есть определенные соображения, и отдавать какие-то программы мы, скорее всего, не будем. Но мы планируем создавать большое количество API-интерфейсов, чтобы разработчики могли интегрировать свои программы с нашими. В частности, подобная система сейчас разрабатывается для контекстной рекламы.

LXF: Какие дополнительные поисковые сервисы вы планируете реализовать в будущем?

АГ: Сервисов планируется выпустить достаточно. Просто у нас пока еще не до всего доходили руки. Например, мы планируем осуществлять поиск и по картин-

кам, и по блогам. Собираемся выдавать пользователю комплексный результат, который будет включать в себя все, что может соответствовать данному понятию – сайты, блоги, карты, изображения и т.д.

LXF: Существует еще один интересный и перспективный вид поиска – поиск по исходным текстам открытого ПО. Будет ли у вас нечто подобное создано?

АГ: Вообще говоря, не составляет никакого труда научить нашу текущую систему искать по программным текстам. Но насколько это нужно, сколько времени потребует доработка и, главное – сколько людей всем этим воспользуется, я не знаю. У нас много предложений по созданию специализированных видов поиска – например поиска по правительственным документам. Как я уже говорил, создание их не представляет никакого труда, но нам пока не хотелось бы распылять свои силы на начальном этапе.

LXF: Любой мало-мальски уважающий себя поисковик рано или поздно обзаводится собственным почтовым сервисом…

АГ: А у нас почта уже есть и сейчас тестируется. Она сделана на AJAX, и мы пока ею пользуемся сами, на себе тестируем. Собственно говоря, создаем единый аккаунт для всех наших сервисов. Он будет необходим, например, для унификации денежных расчетов. Мы планируем создать также собственную внутреннюю платежную систему. Так платить за контекстную рекламу можно будет из одного аккаунта. Сюда же будет входить и еще многое другое. Кроме того, работаем над развлекательными сервисами и идем по абсолютно легальному пути, так как уверены, что в конечном итоге все к этому и придет.

Если представить себе всю структуру Webalta, то поисковик – это как бы некий хребет, который обрастает всем остальным, в данном случае новыми сервисами.

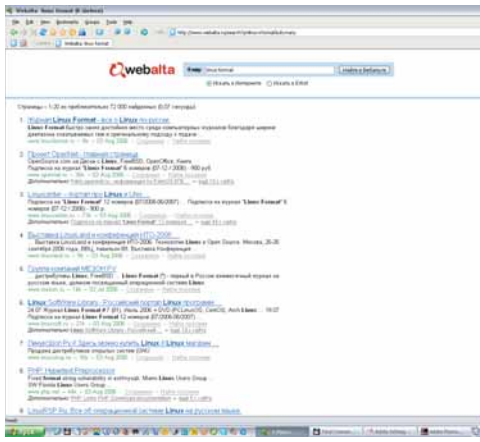
LXF: Для поиска очень большое значение имеет морфология. Морфологический модуль был разработан вашими специалистами или приобретен?

АГ: Да, это приобретение. Мы взяли чужую программу и практически переписали ее. Вообще Webalta в русском и английском языке уже сейчас умеет выделять генетивные пары. Каждый запрос подробно анализируется. Если вводится какой-то сложный запрос например стихи Александра Пушкина в таком-то издании, то система будет искать издание, стихи и имя поэта вместе, и пытаться это обработать. Ведь если еще несколько лет назад люди вводили в строке поисковика всего одно слово, то сегодня уже два-три. Количество информации растет, да и уровень интернет-пользователей постоянно увеличивается. Люди уже научились правильно составлять запросы, и мы видим свою задачу в том, чтобы обрабатывать не просто телеграфный текст или набор ключевых слов, а вполне осмысленные предложения. А для этого нужно научиться разбирать эти предложения и понимать, какие слова важные, а какие нет.

LXF: И последний вопрос Компания Google щедро спонсирует разработчиков браузеров, за что последние зачастую вставляют их в качестве поисковой машины по умолчанию. Были даже курьезные случаи, когда браузер FireFox называли Adware от Google. Собирается ли Webalta оказывать подобную поддержку?

АГ: Я работаю на Маке и пользуюсь браузером Safari. Там по умолчанию установлен только Google, и заменить его невозможно.. Нужно поверх устанавливать специальную программу, которая даст такую возможность. То же самое относится к FireFox. Выставить Webalta в качестве основной поисковой машины там невозможно, нужно писать специальный плагин.

Мы собираемся работать с теми браузерами, которые уже существуют, и будет предоставлять пользователям возможность установить наш поисковик в качестве основного. А что касается спонсирования, то это слишком дорогое удовольствие позволить себе поддержать создание какого-то браузера. Возможно, в будущем это произойдет.





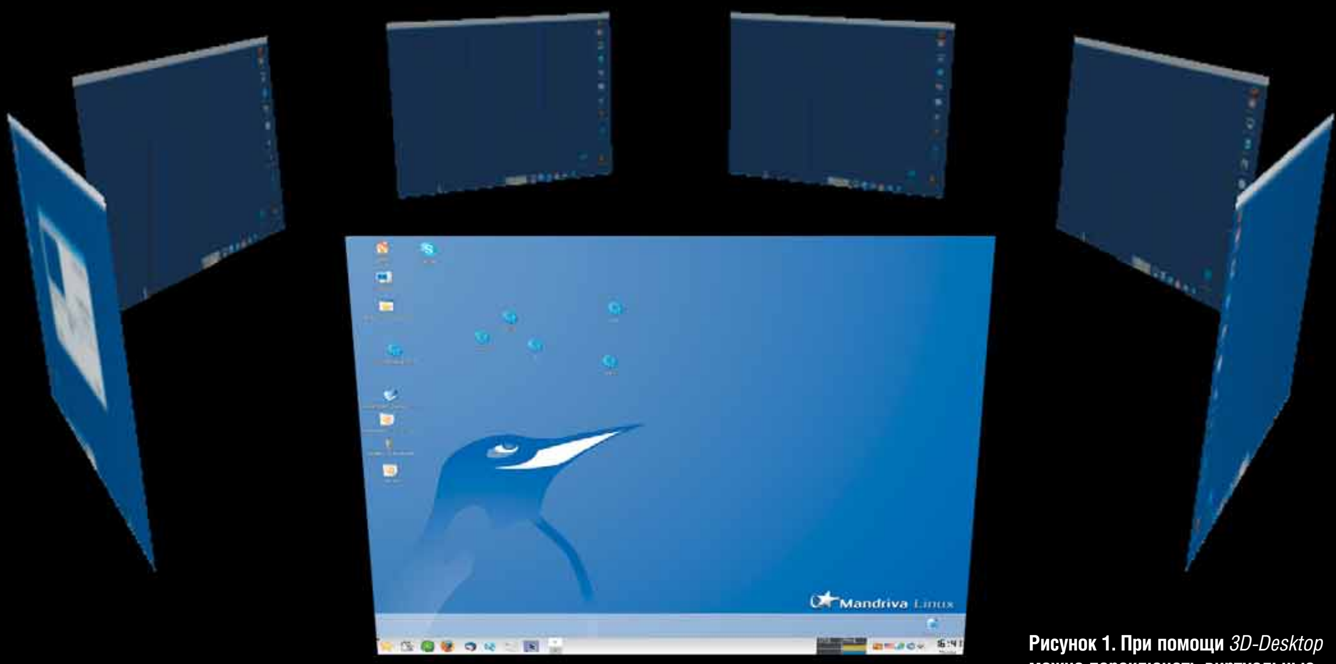


Рисунок 1. При помощи *3D-Desktop* можно переключать виртуальные рабочие столы.

Третье ИЗМЕРЕНИЕ

Часто в фантастических фильмах, рассказывающих о будущем, можно видеть, как пользователь использует при работе с компьютером трехмерный рабочий стол. Но зачем так долго ждать? С будущим можно соприкоснуться уже сегодня, а вашим гидом будет **Сергей Яремчук**.

Операционная система GNU/Linux, построена по принципу *KISS* (Keep It Simple Stupid). В отличие от других ОС, в ней не намешан коктейль из приложений, намертво «вшитых» друг в друга. Философия построения системы, напротив, состоит в использовании небольших по размеру приложений, которые работают независимо друг от друга. При необходимости совместного использования, их всегда можно объединить как непосредственно в командной строке, так и в скриптах при помощи графической оболочки. Кроме того, любой из компонентов заменим на альтернативный, при наличии такового. Графическая подсистема X Window – яркий тому пример. Она не интегрирована в систему, что позволяет вообще ее не устанавливать в случае отсутствия необходимости, а ее сбой никак не сказывается на работе основной системы. Кроме того, существует несколько ее вариантов, наиболее популярные из которых *XFree86* (http://www.xfree86.org/) и *X.Org* (http://www.x.org/). Здесь работает тот же принцип, поэтому X-Window также состоит из нескольких компонентов, а клиент-серверная архитектура позволяет устанавливать их вообще на разных компьютерах. Поэтому и оконных менеджеров и различных расширений в этой системе столько, что вряд ли кто-то возьмется их пересчитать. Каждый проект выражает подход конкретного человека или группы энтузиастов, поэтому нестандартных решений здесь пруд пруди. Если Gnome, KDE, *IceWM* напоминают традиционный рабочий стол, то *Fluxbox* довольно аскетичен по внешнему виду, а разработчики Symphony OS (http://www.symphonyos.com/) вообще целиком и полностью пересмотрели концепцию рабочего окружения пользователя. Естественно, нашлись и такие проекты, которые уже сегодня смотрят в будущее, причем выбор здесь не ограничен известным XGL от Novell. Единственное ограничение- многие из разработок требуют наличия хорошей видеокарты.

Переключатель рабочих столов 3D-Desktop

Разработкой 3D-Desktop (См. врезку 1) занимается фактически один человек: Брэд Вэссон (Brad Wasson). Эта программа, использующая OpenGL/Mesa, позволяет переключать рабочие столы в наглядном 3D-режиме. Выглядит это очень эффектно. После активации пользователю показываются трехмерно расположенные изображения всех виртуальных рабочих столов, которые можно вращать и просматривать. По щелчку на одном из изображений будет

осуществлен переход на выбранный виртуальный стол.

Конструктивно 3D-Desktop состоит из двух программ: демона *3ddeskd* и переключателя *3ddesk*. Демон работает в фоновом и обеспечивает более быструю реакцию, а также кэширование информации об открытых окнах на всех виртуальных столах. Запуск демона лучше произвести отдельно и с опцией **--acquire**, тогда будут собраны изображения со всех рабочих столов.

Дополнительно можно указать время автоматического обновления (в миллисекундах) этой информации.

```
$ 3ddeskd --acquire=1000
===================================
3ddesktop will be acquiring images in one moment. Please wait...
===================================
Daemon started. Run 3ddesk to activate.
```

В некоторых случаях для корректной работы, возможно, потребуется принудительно указать используемый оконный менеджер при помощи опции **--wm**. Поддерживаются следующие варианты: **kde2**, **kde3**, **gnome1**, **gnome2**, **ewmh**, **fluxbox**, **windowmaker**, **enlightenment**, **sawfishonly** и **workspaces**. Задача *3ddesk* – активация сервера, если он еще не запущен и переход в режим переключения рабочих столов. Для компиляции потребуются пакеты *XFree86-devel* или *Imlib2-devel*, либо исходные тексты этих приложений. Для работы желательна аппаратная поддержка 3D-ускорения.

Поддерживаются несколько вариантов размещения и оформления окон. Внешний вид можно скорректировать по своему усмотрению. Так по умолчанию окна выводятся каруселью (carousel) (*Рис. 1*) и при просмотре поворачивается на 180° в горизонтальной плоскости. Но можно указать и другие варианты: **linear**, **cylinder**, **viewmaster**, **priceisright**, **flip**. Выбор варианта показа осуществляется при помощи опции **--mode**.

```
$ 3ddesk --mode=linear
```

Все настройки оформлений сохранены в файле **3ddesktop.conf**, который состоит из нескольких секций описывающих название и параметры вывода. Например:

```
wm kde3
view default
zoom on
show_digit on
digit_size 100
digit_color green
use_breathing false
view bigmoney
mode priceisright
depth 10
digit_color purple
digit_size 150
```

Параметров много и показ рисунков можно очень тонко настроить. Нужная секция выбирается при помощи опции **--view** с указанием имени.

```
$ 3ddesk --view= bigmoney
```

Для удобства запуска *3ddesk* можно создать ярлык на рабочем столе. Например, в KDE для этого достаточно щелкнуть правой кнопкой мыши по пустому месту рабочего стола, выбрать «Создать»->«Ссылку на приложение». В появившемся окне, во вкладке «Общие» нужно указывать название ярлыка, в «Права» дать ему право на выполнение, и в «Приложение» в строке «Команда» записать строку запуска вида **/usr/bin/3ddesk --view= bigmoney**. Теперь по щелчку на ярлыке будет запускаться переключатель рабочих столов, при желании можно указать и комбинацию клавиш.

Трехмерный рабочий стол

Metisse (*Рис. 2*) не является трехмерным рабочим столом в чистом виде. Задачи

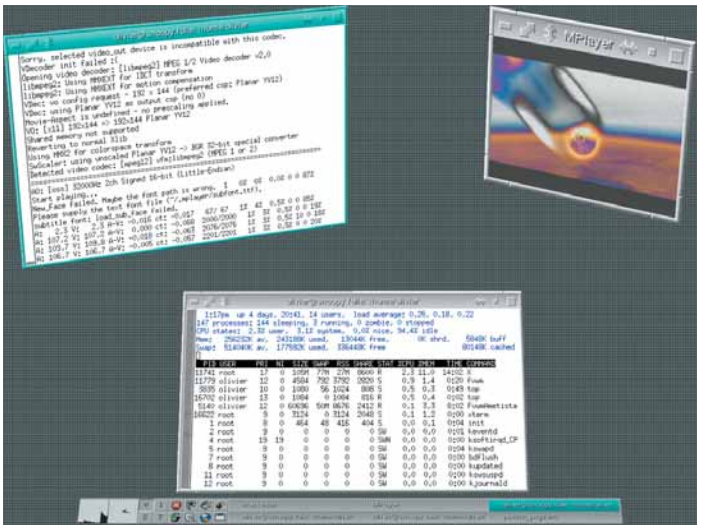


Рисунок 2. ***Metisse*** **обладает всем, что можно представить в трехмерном рабочем столе.**

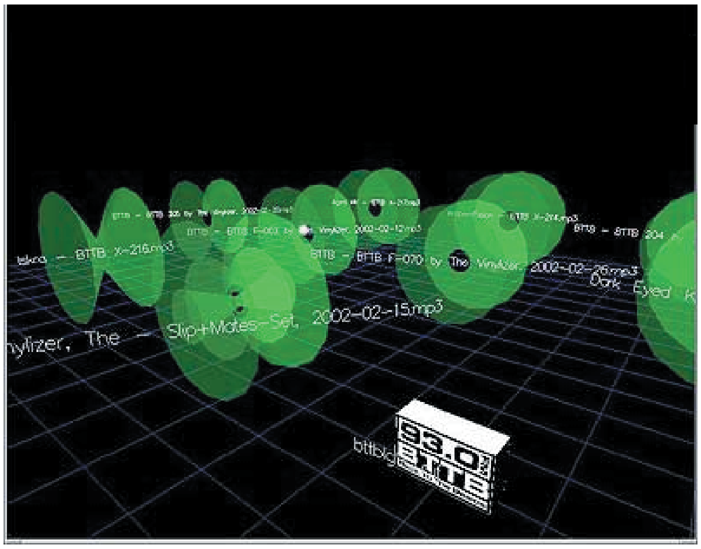


Рисунок 3. Музыкальные файлы в TDFSB представлены в виде дисков.

этого проекта несколько шире и включают разработку и тестирование прогрессивных методов и стандартов. Это, скорее, инструмент для создания новых рабочих сред. Одним из вариантов реализаций такой среды и является 3D. Для установки, кроме самого *Metisse*, вам понадобится библиотека *Nucleo* и *WMI* (Window Manager Icons). Конструктивно Metisse состоит из виртуального X-сервера *Xwnc* (содержащего код **Xvnc** и **XDarwin**), переработанной версии FVWM (исполняемый файл называется fvwmi), и модуля FVWM FvwmAmetista. При помощи меню “Ametista Config” можно настроить некоторое количество параметров, среди которых: изменение прозрачности окон в процессе перемещения, параметры кривизны относительно рабочего стола и его вид, панель, цвет, некоторые реакции на события. Для более глобальных настроек потребуется отредактировать конфигурационный файл. Здесь уже указываются приложения, значки которых будут выводиться в меню, используемые шрифты, количество виртуальных рабочих столов, стиль окна, фоновое изображение. Настройке поддается практически все.

Файловые менеджеры

В GNU/Linux существует большое количество файловых менеджеров, на любой вкус и цвет – от традиционных вроде Midnight Commander, до современных, обладающих большим количеством опций *Konqueror* или *Krusader*. При желании можно добавить в них и третье измерение. Например, *fsv* (3D File System Visualizer) позволяет в более интуитивной и наглядной форме показать пользователю содержимое файловой системы. Для этого он переводит некоторую информацию о файлах и каталогах (например, относительные размеры), которые выводятся затем в 3D виде. На разработку *fsv* студента факультета информатики Массачусетского технологического университета Даниеля Ричарда (Daniel Richard) подтолкнул кадр из фильма Стивена Спилберга «Парк Юрского периода», в котором была показана экспериментальная разработка компании SGI – 3D File System Navigator (http://www.sgi.com/fun/freeware/3d_navigator.html) работающая под *Irix*. В *fsv* каждый показанный объект будет выделен цветом и объемом в соответствии со своими свойствами: размер, тип (каталог, файл, сокет), временем модификации. Для установки потребуется наличие *gtk* и *gtkglarea*.

Другая подобная разработка *TDFSB* – (3D – Filesystem Browser) показывает файловую систему в виде трехмерного мира, по которому можно перемещаться при помощи мыши и клавиатуры. Чтобы зайти внутрь каталога, достаточно щелкнуть по нему мышкой. Музыкальные mp3-файлы показаны как CD (*Рис. 3*), щелчок по «диску» приведет к прослушиванию композиции, аналогичная реакция будет при щелчке по видео-файлу или изображению. Кроме Linux и FreeBSD *TDFSB* портирован также в BeOS и MorphOS. Текущая версия – 0.0.9, но она, несмотря на такой маленький номер, работает стабильно.

Немного по-другому выводит информацию *FileCityMap*. В котором движение по каталогам файловой системы, напоминает путешествие по городу. Скрипты здесь выглядят знаками «Стоп», архивы – мусорными ящиками и т. д. *True3D*Shell* (см. ссылку 11) так же является трехмерным файловым менеджером и чем-то напоминает предыдущую разработку. Пользователь может просматривать как локальные, так и доступные по сети файлы. Кроме того, имеется режим робота, позволяющий автоматизировать просмотр.

Проект *s3d* (см. ссылку 10) представляет трехмерный сетевой дисплейный сервер,

(окончание на стр. 111)

Учебники >>

Наши эксперты помогут вам с любым приложением Linux

ПОЧЕМУ GNU/LINUX



Евгений Балдин переехал на Linux и не думает возвращаться.

«Linux лучше, чем Windows? – Чем лучше-то? – Чем-чем: чем Windows!». Собственно говоря, а действительно чем? Особенно в России. Низкой, точнее, нулевой стоимостью никого не удивишь. Это не является достоинством. Те, кто ожидают увидеть толпу энтузиастов, которые ринутся решать их проблемы за так, то есть на халяву, удивятся услышав, что free значит free as speech (свобода слова), а не free as beer (бесплатное пиво). Собственно говоря, вполне ожидаемо, ведь закон халявы гласит, что её не бывает. Так чем лучше?

И передовых специфичных только для GNU/Linux исследований нигде не ведётся. Информация о научных достижениях доступна всем одинаково. Да и сами реализуемые концепции часто просто повторение уже имевшихся решений. Так чем же лучше?

GNU/Linux – это просто инструмент, который более совершенен, чем другие имеющиеся на сегодня инструменты по причине более продвинутой в социальном плане модели разработки. Модель, которая используется в GNU/Linux, позволяет создавать не просто good enough решения, а решения, совершенные настолько, насколько это возможно в текущей ситуации. Здесь хотя бы есть шанс достичь совершенства.

info@linuxformat.ru

В ЭТОМ ВЫПУСКЕ МЫ ИЗУЧИМ:

СОВЕТ МЕСЯЦА ЧЕРВОТОЧИНА

Скромная команда **mkfifo** едва ли входит в арсенал даже самых заядлых оболочкофилов. Она создает канал для обмена данными или связи двух различных утилит – своеобразную «червоточину» в командной строке Unix. Информация, посылаемая в один конец, может быть считана из другого.

Прежде чем мы приступим к ее практическому использованию, давайте обсудим, где вообще встречаются каналы. Если вы используете оболочку не только для того, чтобы пугать друзей выводом команды **cat /dev/random**, то идея каналов наверняка покажется вам знакомой. Они используются для передачи вывода одной команды на вход другой. Типичный случай – когда команда выводит на экран слишком много текстовой информации. Канал – до **less** или до **more** – позволяет вам получить эти данные в удобочитаемом виде:

```
cat /var/log/messages | less
```

В данном случае, создается временный канал, но ничто не мешает вам создать «постоянное соединение» с помощью **mkfifo**.

Буквосочетание **fifo** в названии команды отражает природу создаваемого канала – «первым пришел – первым ушел» (first in – first out). Создать канал не сложнее, чем набрать **mkfifo <имя_канала>**. Для канала можно также установить права доступа (параметр **--mode**), как для обычного файла.

После того, как канал создан, необходимо обеспечить «закачку» данных. Приведем простой пример. Создадим канал и используем **tail -f** для вывода всех поступающих в него данных:

```
mkfifo fifo_pipe
tail -f fifo_pipe
```

Затем, действуя из другой консоли или учетной записи (если позволяют права доступа), пошлите в канал какие-нибудь данные. Например, наберите **echo "This is a test" >> fifo_pipe** и переключитесь на прежнюю консоль. Вы увидите строку «This is test» в выводе команды **tail**.



ПЕРВЫЕ ШАГИ. СЕРИЯ «LINUX ДЛЯ НОВИЧКОВ»

Рабочие станции: экономим время

Сочетания клавиш, файловые ассоциации и настройка автозагрузки могут сэкономить вам целых... пару минут каждый день. Не отмахивайтесь, утверждает **Энди Ченнел**, одно мгновенье – и вы уже не сможете жить без этих маленьких хитростей...



МЕСЯЦ НАЗАД

Я дал вам несколько советов по безопасной работе с *Firefox*.

Если вы взялись за что-то стоящее, то это также стоит делать быстрее. Или делать как следует. Мы, пользователи Linux, можем разными способами повысить быстродействие компьютера; скажем, пересобрать ядро, докупить более мощный процессор или начать тормозить самим. Но мы не можем обойти вниманием небольшие изменения в настройках, которые экономят всего несколько секунд, но за день, неделю или всю жизнь они сэкономят вам достаточно лишнего времени для игры в *Mario Kart: Double Dash* или загорания.

В этой статье я расскажу вам о небольших приемах, таких как настройка ассоциаций для разных типов файлов, когда за определенный программой закреплен список открываемых по умолчанию типов документов. Мы также установим формат, в котором *OpenOffice.org* сохраняет документы по умолчанию, так что вам не понадобится выбирать его каждый раз. Наконец, можно настроить запуск программ так, что любимое приложение будет запускаться без дополнительных усилий. Эти хитрости дают заметный результат и повышают ваши навыки владения компьютером. Я считаю, что через некоторое время вы заметите отдачу и начнете пользоваться ими ежедневно.

ЧАСТЬ 1 – ОТКРЫВАЕМ ДОКУМЕНТЫ

Разработчики KDE и GNOME любезно составили список типов файлов и связанных с ними приложений, которые мы используем в Linux. Однако бывают случаи, когда вас не устраивают стандартные настройки. На наше счастье, обе графические среды позволяют внести нужные исправления.

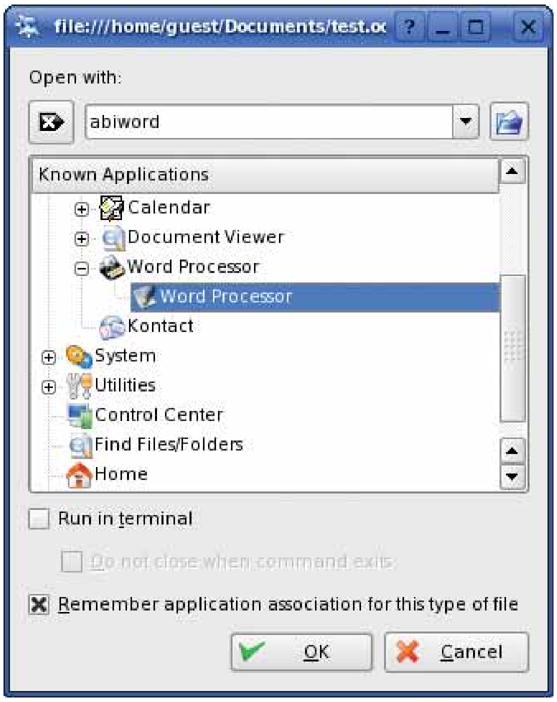
Начнем с KDE. В этом примере у меня есть свежая установка SUSE Linux с KDE 3.5, но какой-то умник забыл поставить во время установки офисные программы. Установка *AbiWord* решает эту проблему, и теперь я вновь могу писать статью для LXF. Тем не менее, так как изначально в SUSE не было этой программы, для файлов документов не были установлены и нужные ассоциации. Это легко проверить.

Найдите какой-нибудь документ; мне, к примеру, нужно открыть файл *OOo Writer*. Родные документы *AbiWord* – с расширениями .abw и .abt – открываются простым двойным щелчком, однако другие типы документов, включая .odt, еще не «привязаны» к нужной программе. Чтобы исправить это, щелкнете правой кнопкой мыши по файлу и выберите «Открыть в > Другое приложение», после чего появится диалог выбора программ. В основной части окна будет список программ, соответствующий структуре K-меню в KDE. Найдите группу Офис, щелкнете по маленькой стрелке напротив имени группы и в появившемся списке выберите *AbiWord*. В зависимости от настроек KDE, программа может называться «Word Processor» (для того чтобы изменить отображаемое имя, смотрите врезку «Быстрые подсказки» на последней странице статьи). По нажатию кнопки OK документ будет открыт для редактирования в *AbiWord*.

Но не торопитесь! Нам нужно убедиться в том, что этот тип документа всегда открывается в выбранном приложении. Поставьте флажок в нижней части окна на пункте «Запомнить связь с приложением для этого файла» и нажмите OK. Только теперь *AbiWord* станет стандартным приложением для этого типа файлов.

Если вы закроете теперь *AbiWord* и щелкнете по файлу правой кнопкой мыши еще раз, вы заметите, что контекстно-зависимое меню «Открыть с помощью» пополнилось новым пунктом – «Открыть с помощью *AbiWord*/Word processor». Так как этот пункт выбран по умолчанию, при двойном щелчке по файлу будет запускаться именно *AbiWord*.

В какой-то момент вам может захотеться изменить привязку файлов; например, если вы установили *OpenOffice.org* и пристрастились к работе с odt-файлами именно в нем.



Поставьте флажок, и *AbiWord* будет всегда запускаться при щелчке по odt-документам

В таком случае, повторите описанную выше процедуру, выбрав другую программу и не забыв отметить флажком постоянную привязку. Теперь документы будут открываться уже в новой программе.

Как это делается в GNOME

Учитывая схожесть разных графических оболочек, не удивляйтесь, что это действие будет выглядеть почти также в GNOME, хотя кое-что здесь реализовано более удобно (например, когда для открытия файла доступны два приложения, правый щелчок мыши позволит вам выбрать сразу же одно из них), а кое-что – неуклюже (настроить приложение по умолчанию здесь не так просто, как в KDE).

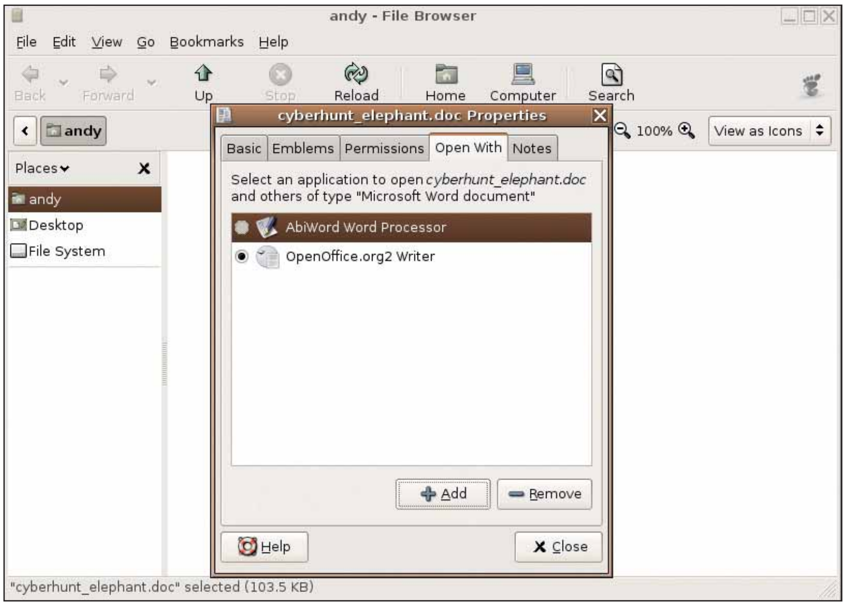
Приведем пример: допустим, что SUSE мне надоел и я решил попробовать Ubuntu. По умолчанию в состав Ubuntu Dapper Drake входит Gnome 2.14 и *OpenOffice.org* 2.0, который призван открывать разнообразные офисные документы. Тем не менее, привыкнув к *AbiWord*, я хочу сделать так, чтобы после его установки я мог бы щелкнуть в *Nautilus* правой кнопкой мыши по файлу и выбрать эту программу. Обратите внимание, что по двойному щелчку мой файл откроется в OOo, поэтому мне нужно именно контекстное меню с выбором приложений.

Если мы откроем файл в *AbiWord* через контекстное меню, этот выбор не будет сохранен и файл откроется в данном приложении только в этот раз. Для того, чтобы назначить приложение по умолчанию, зайдите в свойства файла (Options) и перейдите на вкладку «Открывать в...» (Open with...). Вы увидите список тех приложений, которые ассоциированы с данным типом файлов. В моем случае для файлов с расширением .doc имелось два приложения – *AbiWord* и OOo. Отметив нужный пункт при помощи кнопки-переключателя, я могу задать стандартную программу для открытия документов.

Кроме этого, у вас имеется возможность редактировать состав списка программ при помощи кнопок «Добавить» (Add) и «Удалить» (Remove), которые располагаются в нижней части диалогового окна. Это может понадобиться в том случае, когда вы установили скачанную программу для просмотра графики вручную (а не через менеджер пакетов Ubuntu) и захотели сделать ее стандартной для png-файлов. Для этого вам вначале понадобится включить эту программу в список стандартных приложений для графических файлов. Это можно сделать и через контекстное меню, вызываемое при правом щелчке мыши, выбрав пункт «Open With Other Application» (Открыть в другом приложении). Найдите приложение в списке, или укажите путь к исполняемому файлу в браузере – теперь программа доступна в числе стандартных приложений. Прекрасно.

В KDE реализован альтернативный способ редактирования файловых ассоциаций. Для этого можно открыть Центр управления и найти пункт «Компоненты KDE > Привязки файлов». Этот способ отнимет у вас чуть больше сил, потому что вам придется «прочесывать» весь гигантский список типов файлов, сгруппированных по тематическому принципу – Звук, Изображения, Текст. Найдите нужный тип и задайте команду, которая будет выполняться при двойном щелчке по файлу.

Это делается так. Отыщите нужный шаблон расширения файла в списке или добавьте его вручную при помощи кнопки «Добавить» (Add). Затем найдите в правой части окна раздел Application Preference Order (Порядок запускаемых приложений), нажмите кнопку «Добавить» (Add) и введите команду, которая запустит нужную программу. Добавленное приложение можно двигать вверх или вниз, меняя его положение в списке.



Диалог выбора программ в GNOME опрятен и ничем не перегружен.

ЧАСТЬ 2 – ЗАПУСКАЕМ ПРОГРАММЫ

Большинство людей пользуется одними и теми же приложениями изо дня в день, поэтому было бы логичнее запускать их автоматически при утренней загрузке компьютера, когда вы готовите себе кофе.

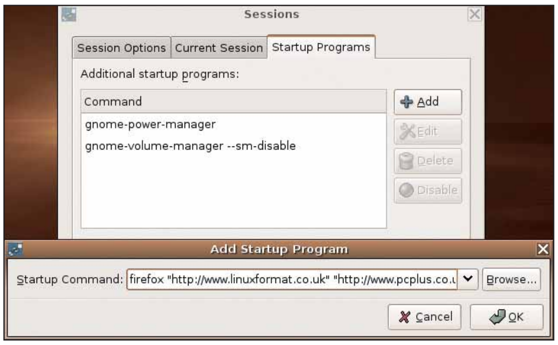
В KDE вам нужно будет отыскать папку **Autostart**, куда помещаются все ссылки на автозагрузку. В большинстве дистрибутивов эта папка имеет адрес **/home/имя_пользователя/.kde/Autostart**. Вы наверно уже заметили, что в вашей домашней директории не видно папки с именем **.kde**. Это происходит из-за точки (.), стоящей в начале имени и означающей, что данная папка является скрытой. Вам понадобится включить отображение скрытых файлов, выбрав пункт View > Show Hidden Files (Вид > Показывать скрытые файлы) или нажав Ctrl+H когда в *Konqueror* открыта ваша домашняя папка.

Любое приложение или ссылка, помещенные в папку автозагрузки, будут автоматически запускаться при старте KDE, поэтому вам не составит труда щелкнув по любому месту в папке правой кнопкой мыши, выбрать Create New > Link To Application (Создать > Ссылку на приложение) и ввести команду запуска нужной программы. Если у вас есть привычка посещать определенный сайт в начале работы, вы также можете создать ссылку на URL сайта, выбрав пункт Create New > Link To Location (Создать > Ссылку на местоположение). Таким >>

образом *Konqueror* будет открывать нужный вам сайт при загрузке компьютера.

Если по какой-то причине папки **.kde/Autostart** нет в вашей системе, это означает, что ее расположение было изменено в настройках путей. Не беспокойтесь, все можно поправить. Откройте Центр управления и перейдите в раздел System Administration > Paths (Системное администрирование > Пути). Здесь вы можете указать путь к папке автозагрузки или хотя бы выяснить, где она находится в данный момент. Если вам захочется упростить доступ к автозагрузке (и избавиться от возни со скрытыми папками), вам достаточно создать ее в вашей домашней директории, к примеру, под именем Start Up, и прописать путь к ней в указанном разделе Центра управления. Теперь вы можете просто перетаскивать ссылки из К-меню в окно автозагрузки и в появившемся окне выбора указать Link To Application (Ссылка на приложение).

При следующем запуске KDE ваши программы из папки автозагрузки будут запущены автоматически [имейте в виду: каждое такое приложение увеличивает время запуска системы, – прим. ред.].



Автозагрузка в GNOME может больше, чем кажется. Эта команда запустит *Firefox* с двумя сайтами в разных вкладках.

Автоматический переход на web-адреса

Автозагрузка в GNOME реализована в виде небольшой программы, а не папки, и вам будет несложно в ней разобраться. Эта утилита находится в System > Preferences > Sessions (Система > Параметры > Сеансы) на вкладке Start Up (Запуск при старте). Для того, чтобы добавить в список, к примеру, **Firefox**, нажмите кнопку Add (Добавить) и наберите в появившемся окне команду *firefox*. Самое приятное состоит в том, что вы можете указывать здесь совершенно любые команды. Если вы, например, хотите, чтобы *Firefox* открывал сайт *LXF* при каждой загрузке GNOME, вам понадобится всего одна команда:

firefox "http://www.linuxformat.ru".

Более того, вы можете автоматически запускать *Firefox* с несколькими вкладками, указав несколько web-адресов подряд, каждый в отдельных кавычках. Да, и не забудьте поставить одинарный пробел между закрывающей кавычкой одного и открывающей кавычкой другого сайта. Следующая команда откроет *Firefox* с сайтом *LXF* Russia в первой вкладке, и *LXF* UK во второй:

```
Firefox "http://www.linuxformat.ru" "
http://www.linuxformat.co.uk"
```

Полезные сочетания клавиш

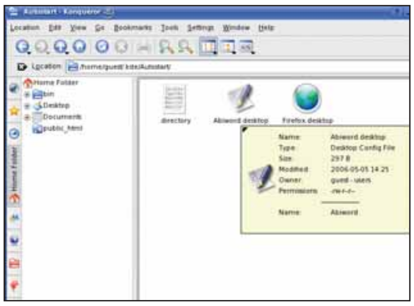
Вы можете запускать программы мгновенно, если привяжите их к определенным клавиатурным комбинациям. Но будьте внимательны: не используйте такие сочетания клавиш, которые уже назначены действиям или командам в ОС. Например, если вы настроите запуск *AbiWord* при нажатии Ctrl+C, то это заблокирует копирование выделенного фрагмента в буфер обмена. Более безопасными сочетаниями являются комбинации Alt/Ctrl с цифровыми клавишами, поэтому если вы обычно используете три программы – скажем, текстовый процессор, браузер и почтовую программу, то есть смысл назначить им легко запоминаемые комбинации Ctrl+1, +2 и +3 соответственно.

В KDE откройте Центр управления и перейдите в раздел Desktop > Panels > Menu (Рабочий стол > Панели > Меню). Щелкнете по кнопке Edit K Menu (Редактировать К-меню) в низу раздела и найдите настройки клавиатурных комбинаций для запуска программ. После того, как вы все настроите, не забудьте сохраниться (File > Save), иначе изменения будут потеряны.

Редактировать меню в GNOME вы можете при помощи утилиты *Alacarte Menu Editor*, которую вы найдете в разделе Applications > Accessories (Приложения > Аксессуары). Однако разработчики похоже не спешат разрешать пользователям создавать свои клавиатурные сокращения. Конечно, это не значит, что их вовсе нельзя создавать, просто нужно немного поусердствовать.

Для того, чтобы настроить запуск *AbiWord* по нажатию Ctrl+1, нам понадобится Редактор Gconf (Gconf-Editor). Здесь вы можете поиграть с системными настройками GNOME, поэтому будьте осторожны. Откройте терминал (Application > Accessories > Terminal) и наберите **gconf-editor**, после чего откроется окно редактора *Gconf*. Используйте небольшие стрелки напротив разделов и разверните их дерево до пункта Apps > Metacity > Keybinding_command, после чего щелкнете два раза по Command_1. В качестве значения укажите команду, которая запускает вашу программу (ее можно выяснить, щелкнув правой кнопкой мыши по ярлыку программы в *Alacarte Menu Editor*), и нажмите OK.

Не выходя из ветки Metacity, выделите пункт Global_keybindings и найдите в списке ключ Run_command_1. Щелкнете по нему два раза и в появившемся окне задайте комбинацию клавиш. Для Ctrl+1 необходимо набрать **<Control>1** именно так, в треугольных скобках, для того, чтобы компьютер воспринял это именно как специальную клавишу, а не слово "Control". Теперь можно нажать OK, закрыть *GConf* и проверить работу новой «горячей клавиши».



Скопируйте приложения сюда для их автозапуска...



... или сделайте любую папку в вашем компьютере местом для автозагрузки.

ЧАСТЬ 3 – СОХРАНЯЕМ ДОКУМЕНТЫ

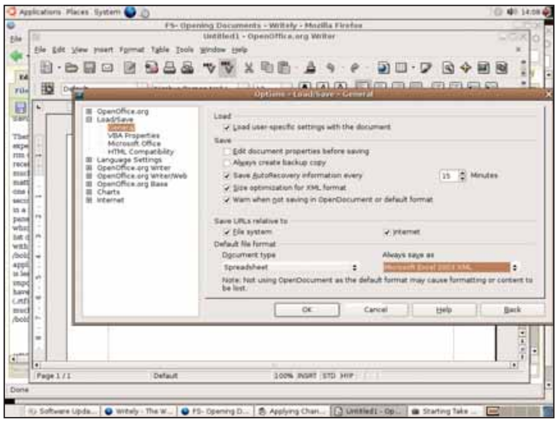
Нет сомнений, что *OpenOffice.org* входит в зенит своей славы по мере того, как все большее число людей рассматривают его в роли альтернативы дорогому MS Office, пользователи которого также обеспокоены моральными и правовыми проблемами, связанными с пиратством. Но и в нашей бочке отборного меда закралась ложка дегтя: пока что пользователей *OOo* еще слишком мало, и пересылаемые по почте документы часто приходится открывать сторонними программами. Для пользователя *OOo* файл в «неродном» формате обычно не вызывает никаких проблем, так как офисный пакет имеет хорошие фильтры для обработки .doc, .xls, .ppt и других закрытых форматов. Другое дело с документами, созданными в *OOo*. Вы должны осознавать, что пользователь, который получит ваш ods-файл, может вообще не понять, что это за спам он получил.

Решение этой проблемы – выбрать более универсальный открытый формат или, стиснув зубы, пойти «в массы» с проприетарным форматом. Для того, чтобы сэкономить время и не мучаться с выбором фор-

мата при каждом сохранении документа, мы можем заранее определить для каждого приложения тот формат, который будет использоваться по умолчанию.

Начните с окна настроек Tools > Options (Инструменты > Настройки) и перейдите в раздел Load/Save – General (Загрузка/Сохранение – Основные). В нижней части окна вы заметите пару выпадающих списков для выбора формата файлов по умолчанию. Список слева предлагает выбрать тип документа – текст, таблица, презентация и так далее, в то время как список справа является контекстно-зависимым и предлагает вам ряд соответствующих форматов.

Если вы плотно общаетесь с пользователями *Excel*, то для электронных таблиц есть смысл задать формат Microsoft Excel 2003 XML, который убережёт вас от проблем совместимости с коллегами.



***OOo* может автоматически сохранять файлы в разнообразных форматах.**

Выбираем формат файла

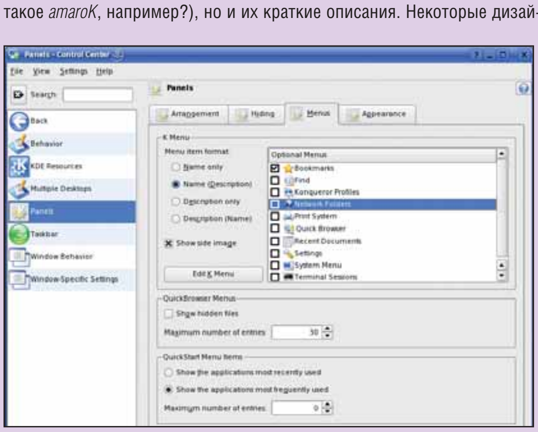
Выбрать правильный формат для ваших документов очень важно. Варианты, о которых речь пойдет ниже, отражают моё личное мнение, однако я постарался предложить альтернативы закрытым форматам везде, где это возможно. Тем не менее, иногда нам приходится отступать от своих принципов и использовать проприетарные форматы в тех случаях, когда это требуется формой документа или, скажем, ВУЗом.

- **Текстовые документы** *OOo* и *AbiWord* неплохо открывают doc-файлы, чего нельзя сказать о пакете *KOffice*, где все еще далеко от идеала. Программы, работающие через web, такие как *Writely* (**www.writely.com**) и *ThinkFree Office* (**www.thinkfree.com**) тоже умеют открывать и сохранять файлы в формате DOC вполне прилично. В случае, когда необходимо гарантировать одинаковый внешний вид документа на разных компьютерах и заказчику не понадобится править файл, выбирайте PDF. Тем не менее, наиболее универсальным выбором будет Rich Text Format (.rtf), который поддерживает несложные таблицы, цветовое выделение, различные начертания шрифтов (полужирное, курсив) и выравнивание текста. Перечисленные форматы открываются в большинстве текстовых процессоров и имеют обычно меньший размер, чем файлы других форматов.
- **Электронные таблицы** Для тех счетоводов, которым важно наладить между собой общение, у нас есть пара форматов на выбор. Первый предлагает вам воспользоваться одной из версий документа *Microsoft Excel*, в то время как второй представляет собой текстовой вариант CSV(Comma-Separated Values, значения, разделенные запятыми). Последний вариант хорошо годится для простых числовых таблиц, но как только ваш документ начнет усложняться – а вы ведь не можете заставить коллег перейти на открытые программы – лучшим выбором оказывается формат *Excel*.
- **Презентации** В *OOo* вы найдете всего два формата для презентаций: *Impress* или *Powerpoint*. Если вы не уверены в том, что на каждом компьютере, где вы будете открывать свою презентацию, установлен *OpenOffice.org*, используйте формат PPT. Тем не менее, если вам этот формат вообще не нужен, не забывайте, что в *Impress* имеется впечатляющий набор фильтров для экспорта в HTML, Flash, PDF и стандартные графические форматы. При экспорте в статические форматы, такие как JPEG вы потеряете эффекты перехода между слайдами, но зато сами слайды будут гарантированно верно отображаться на любом компьютере [при этом слайды в формате JPEG могут выглядеть «неряшливо», так как он предполагает сжатие с потерями, – прим.ред.].
- **Фотоснимки** Фотографии не вызывают таких проблем, как другие типы документов из-за того, что вся отрасль цифровых камер и компьютеров договорилась о поддержке JPEG. Для лучшего качества вы также можете использовать форматы PNG и TIFF, хотя tif-файлы будут очень большими по размеру.
- **Рисунки и иллюстрации** Большая часть приложений для работы с векторной графикой стандартно использует формат EPS, разработанный компанией Adobe. Но времена меняются, и многие программы теперь поддерживают более современный стандарт SVG. Если вы пересылаете векторный рисунок для дальнейшей правки – нет ничего лучше EPS. Если же правка не требуется, вполне подойдёт и PDF.

Как и большинство вещей в мире Linux, приложение и формат файлов, с которыми вы работаете, зависят от того, какие правила игры вы установите и с кем будете иметь дело. Конечно, идеи и соображения в этой статье не приведут к мировым волнениям, но зато сэкономленные несколько минут позволили мне успеть проиграть в *Mario Kart*. LXF

БЫСТРЫЕ ПОДСКАЗКИ

Многие поставщики KDE-дистрибутивов стараются, ради нашего с вами удобства, сделать так, чтобы в К-меню отображались не просто названия программ, которые могут ничего не говорить пользователю (что такое *amaroK*, например?), но и их краткие описания. Некоторые дизайнеры интерфейсов вообще оставляют одни описания. Этот подход хорош до тех пор, пока у вас один текстовый процессор, один медиаплейер и так далее, но когда вы устанавливаете дополнительные приложения со схожими функциями, описания становятся проблемой. Но выход есть!



KDE позволяет настроить главное меню на свой вкус.

Найдите в К-меню Центр управления (у него, к счастью, имя и описание совпадают) и перейдите в раздел Рабочий стол > Панели. В горизонтальном ряду вкладок выберите «Меню». В появившемся разделе будет много настроек, связанных с удобством использования, поэтому просмотрите их внимательно. В главном окне, к примеру, можно настроить быстрый доступ к закладкам в Konqueror, к недавним документам и сетевым папкам. В нижней части окна можно настроить поведение К-меню, которое может запоминать либо наиболее часто запускаемые программы, либо недавно запущенные.

Нас, тем временем, интересует раздел Menu Item Format (Формат элемента меню). Здесь вам предлагается четыре варианта: первые два отдают приоритет названию программы (идеально, если вы ориентируетесь в них), а другие два – её описанию. Вы заметите, что для каждой программы уже имеются как название, так и описание; окно настройки лишь переключает отображаемые в меню элементы.

ЧЕРЕЗ МЕСЯЦ

Дети и компьютеры: где найти панацею? Мы посмотрим, как обезопасить ребят и технику друг от друга.



ПРАКТИЧЕСКИЕ ЗАНЯТИЯ ВЕКТОРНАЯ ГРАФИКА

Inkscape Создание визитки

ЧАСТЬ 2 **Дмитрий Кирсанов** представляет простой, но поучительный дизайнерский проект – создание визитной карточки!

МЕСЯЦ НАЗАД



Мы учились рисовать при помощи инструмента «Каллиграфическое перо» (Calligraphic pen).

При всем многообразии электронных способов коммуникации, изобретенных за последние 20 лет, визитная карточка по-прежнему популярна. Это – ваше лицо, послание в бутылке, маленькая презентация того, что вы сами считаете главным в себе. Не удивительно, что оформлению крошечного картонного квадратика уделяется так много внимания. Как в японском трехстишии, здесь у вас крайне ограниченное пространство для самовыражения.

Учитывая все это – простоту, ограниченное пространство и необходимость стильного оформления, – создание визитной карточки может стать превосходным экзаменом для векторного редактора. *Inkscape* подойдет для этого как нельзя лучше. Единственное его слабое место – это получение готовых к печати выходных файлов для сложных изображений. Но и здесь есть некоторые обходные пути, которые мы с вами изучим на этом занятии.

Inkscape пока не может соревноваться с лучшими коммерческими редакторами вроде *Adobe Illustrator* или *Corel Draw* по количеству команд и инструментов. Но в повседневной дизайнерской работе основное время уходит не на отладку цветовых профилей или рисование замысловатыми кистями, а на куда более простые вещи – перемещение объектов, масштабирование, пробу различных шрифтов и расцветок и т. д. Здесь-то и покажет себя *Inkscape* – со своим ненавязчивым интерфейсом, многочисленными клавиатурными комбинациями и общей предупредительностью.

Стань художником

Предлагаемое руководство описывает два разных способа оформления карточек, но я не предлагаю вам идти за мной след в след (разве что для изучения технических тонкостей). Если вы планируете создать реальную визитную карточку, начните с того, чтобы найти и просмотреть как можно больше примеров хорошо сделанных карточек. Затем откройте Inkscape и поиграйте с формами, шрифтами и цветами, стараясь соединить лучшее из найденного с вашими собственными идеями.

Я дизайнер, так что мои примеры могут показаться вам несколько экстравагантными – вероятно, ваша карточка будет выглядеть более традиционно. Тем не менее, и эти примеры могут быть поучительны, если наблюдать весь процесс от начала до конца. Итак, приступим.



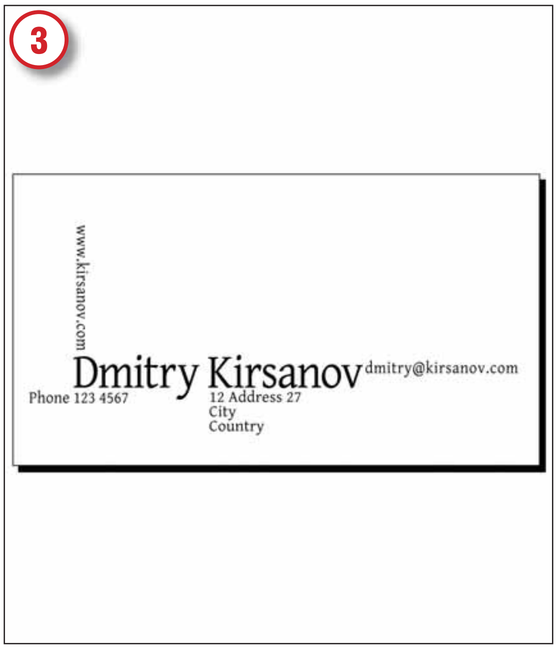


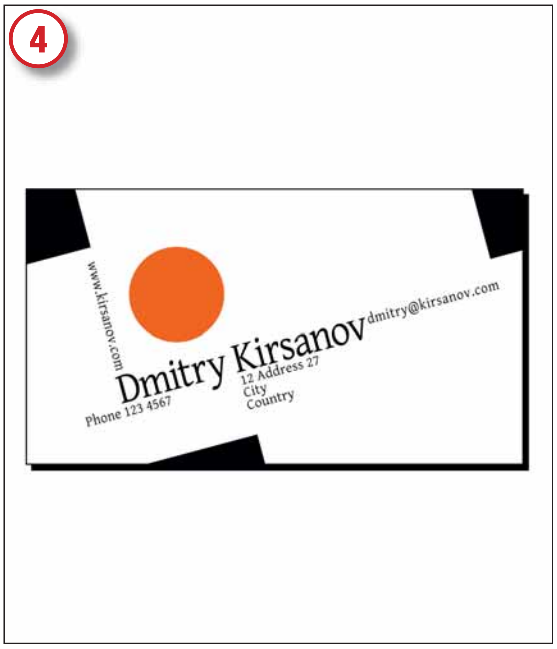
Первые шаги

Создайте новый документ, выбрав шаблон под названием «Business Card 90x50mm» в меню File > New (Файл > Создать) (A). Если вам необходим другой размер, то его всегда можно изменить в диалоговом окне Document Preferences (Параметры документа) – Ctrl+Shift+D. Затем переключитесь в инструмент Текст и создайте текстовые объекты для каждого элемента – имя, должность, адрес, номер телефона и т.д. Все они должны быть независимыми объектами (щелкните и наберите текст для каждой строки отдельно), потому что нам придется двигать и трансформировать их.

Выбираем шрифт

Следующий логический шаг – выбор шрифта (шрифтов) для ваших текстовых объектов. Выбранный шрифт сразу задаст общий стиль композиции. После выбора шрифта для всех текстовых строк (A) попробуйте изменить их относительные размеры (B) инструментом Selector. Смотрятся ли рядом строки одного шрифта, но разных размеров? Если нет, можно попробовать разные шрифты (но в любом случае не стоит использовать больше двух шрифтов на одной карточке).





Выравниваем элементы

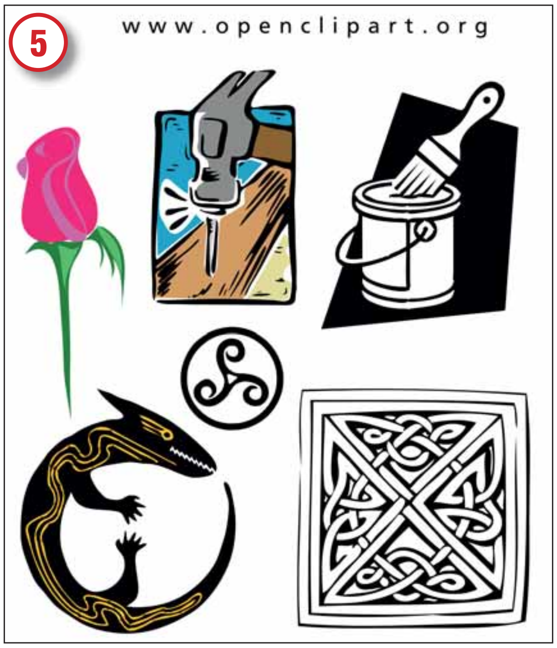
Создание визитной карточки для одного человека (а не шаблона для многих карточек) хорошо тем, что можно размещать и выравнивать текстовые объекты точно, не оставляя дополнительного места для имен и адресов разной длины. Я разложил компоненты адреса вокруг своего имени, выровняв их по нескольким невидимым линиям. Получилась асимметричная, но тесно увязанная композиция.

Конструктивизм

Давайте попробуем теперь выделить все (Ctrl+A) и немного повернуть (один раз нажав [). Намного лучше! Что же это мне напоминает? Здесь определенно есть что-то от конструктивизма – недолговечного, но влиятельного стиля, который был популярен в 1920-х годах. Конструктивисты любили сыграть на контрасте шрифтов и пустить текст под углом. Давайте завершим композицию тремя черными уголками и большим красным кругом в центре: конструктивисты любили простые геометрические фигуры и сочетание черного с красным.

>>

<<





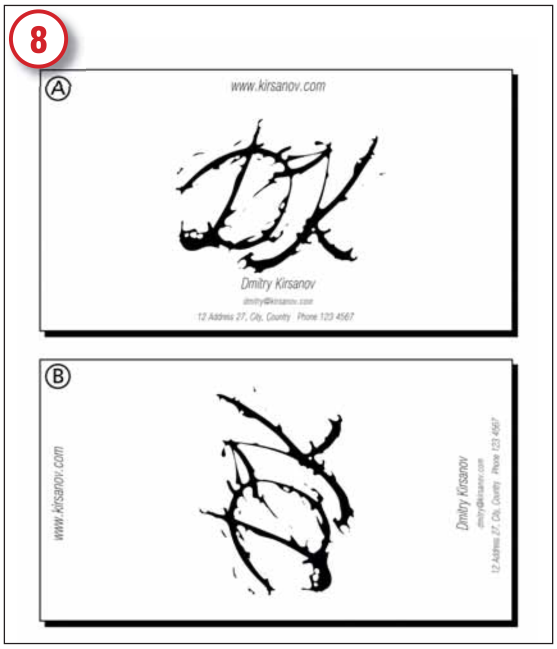
Другие источники вдохновения

Наша первая визитная карточка в основном готова. Так получилось, что ее оформление состоит по большей части из текстовых строк, а графические элементы послужили лишь дополнением к основному замыслу. Но это не единственный возможный подход. Для создания корпоративной визитной карточки естественной отправной точкой станет логотип компании. Если вы хотите сделать карточку более индивидуальной, попробуйте оттрассировать свою фотографию (Path > Trace Bitmap). Бесплатную графику можно также найти на **http://openclipart.org**.

Иной подход

Для второй карточки попробуем совсем другой подход: стилизованные инициалы. Инструментом Каллиграфическое перо (угол 90, фиксация 1.0) я нарисовал вензель из букв «D» и «K». Когда мне наконец удалось придать буквам более-менее правильные очертания, результат выглядел неплохо, но довольно топорно (A). Чтобы пригладить рисунок, объединим все штрихи в один контур (Ctrl+K) и несколько раз применим упрощение (Ctrl+L), втяжку и растяжку (Alt+), Alt+() (B).



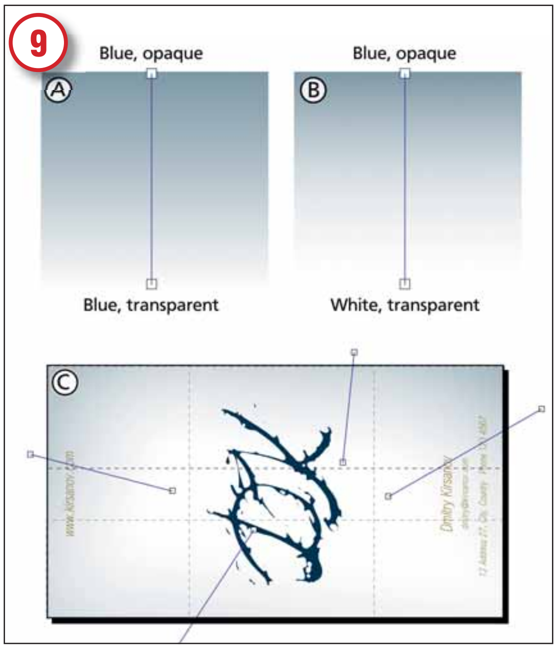


Взлохмачиваем буквы

Теперь буквы выглядят более естественно – но их можно сделать еще интереснее. Я взял более тонкое перо, до отказа увеличил параметр Tremor (Дрожь) – это новая функция в *Inkscape 0.44* – и от души порезвился вокруг вензеля своим планшетным пером. Первый результат выглядел не очень вдохновляюще (A) – но только потому, что я забыл проделать обычные магические пассы «упрощение – втяжка – растяжка» (B).

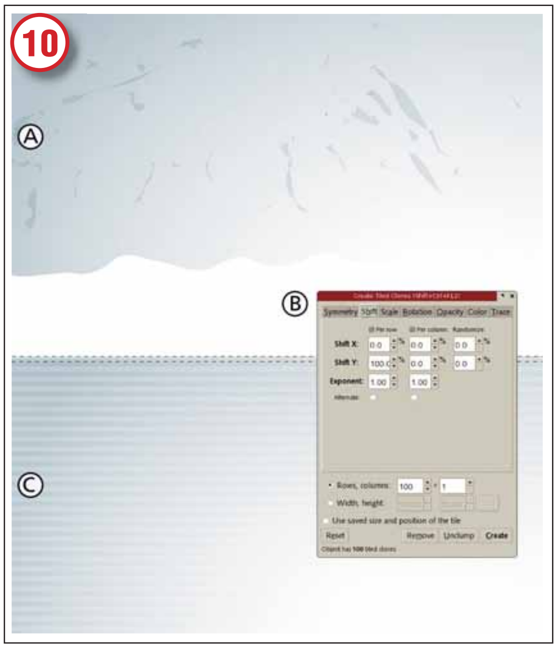
Рассаживаемся по местам

Пожалуй, это произведение заслуживает центрального места на карточке, а все остальное можно симметрично расположить вокруг него (A). Я взял простой курсивный, очень легкий шрифт без засечек, чтобы он не отвлекал внимания от вензеля в центре. Однако горизонтальное положение в данном случае не лучший вариант – тексту в нем тесно. Поворот на 90 градусов (Ctrl+[) дает больше места для вензеля, если отбить имя и адрес к самому краю (B).



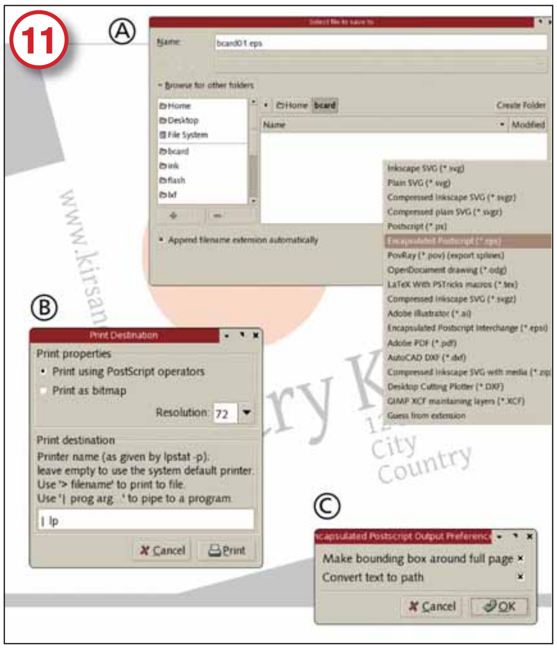
Добавляем градиенты

Теперь попробуем добавить фоновый градиент. Стандартный градиент от непрозрачности к прозрачности некоторого цвета (например, синего, A) выглядит довольно грубо. Но есть одна уловка, значительно улучшающая вид градиента на белом фоне: сделайте прозрачную сторону градиента белой, а не синей (B). На (C) видно, как я добавил четыре несимметричных прямоугольника с градиентами по краям карточки для придания ей мягкой, естественной выпуклости.



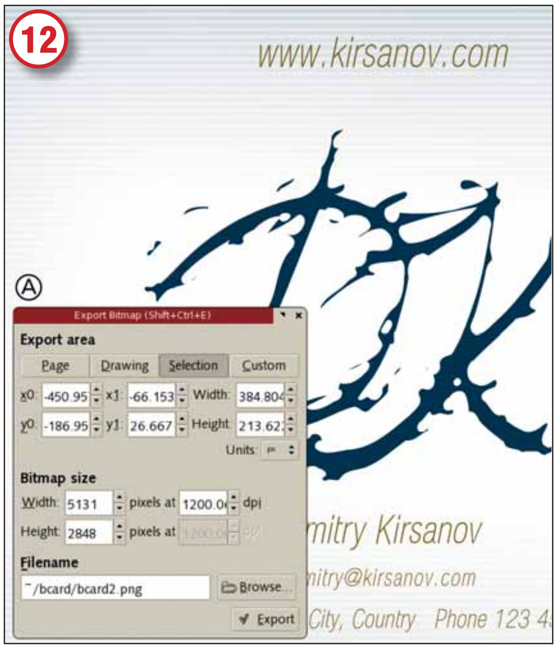
Добавляем текстуру

Можно сделать карточку еще интереснее, разбросав по ней маленькие полупрозрачные каллиграфические штрихи (A, увеличено). Другой прием – сетка из полупрозрачных линий. Начертите узкий горизонтальный прямоугольник белого цвета, затем откройте диалоговое окно Tile Clones (Расположение клонов, B) и создайте узор с симметрией P1, 100 рядов в 1 колонку, с 100% вертикальным сдвигом между рядами. Сгруппируйте прямоугольники и поместите их поверх градиента, но под вензелем и текстом. Прозрачность отрегулируйте по вкусу (C).



EPS экспорт

Теперь у нас есть два образца визитных карточек – но как же распечатать их? Конструктивистский вариант не содержит ни градиентов, ни прозрачности; это просто набор плоских непрозрачных форм, поэтому он может быть без потерь сохранен в форматах PS, EPS (A) или прямо отпечатан на PostScript-принтере (B). В диалоговом окне параметров EPS (C) включите параметры Convert Text To Path (Конвертировать текст в линии) и Make Bounding Box Around Full Page (Ограничить область пределами страницы).



Растровый экспорт

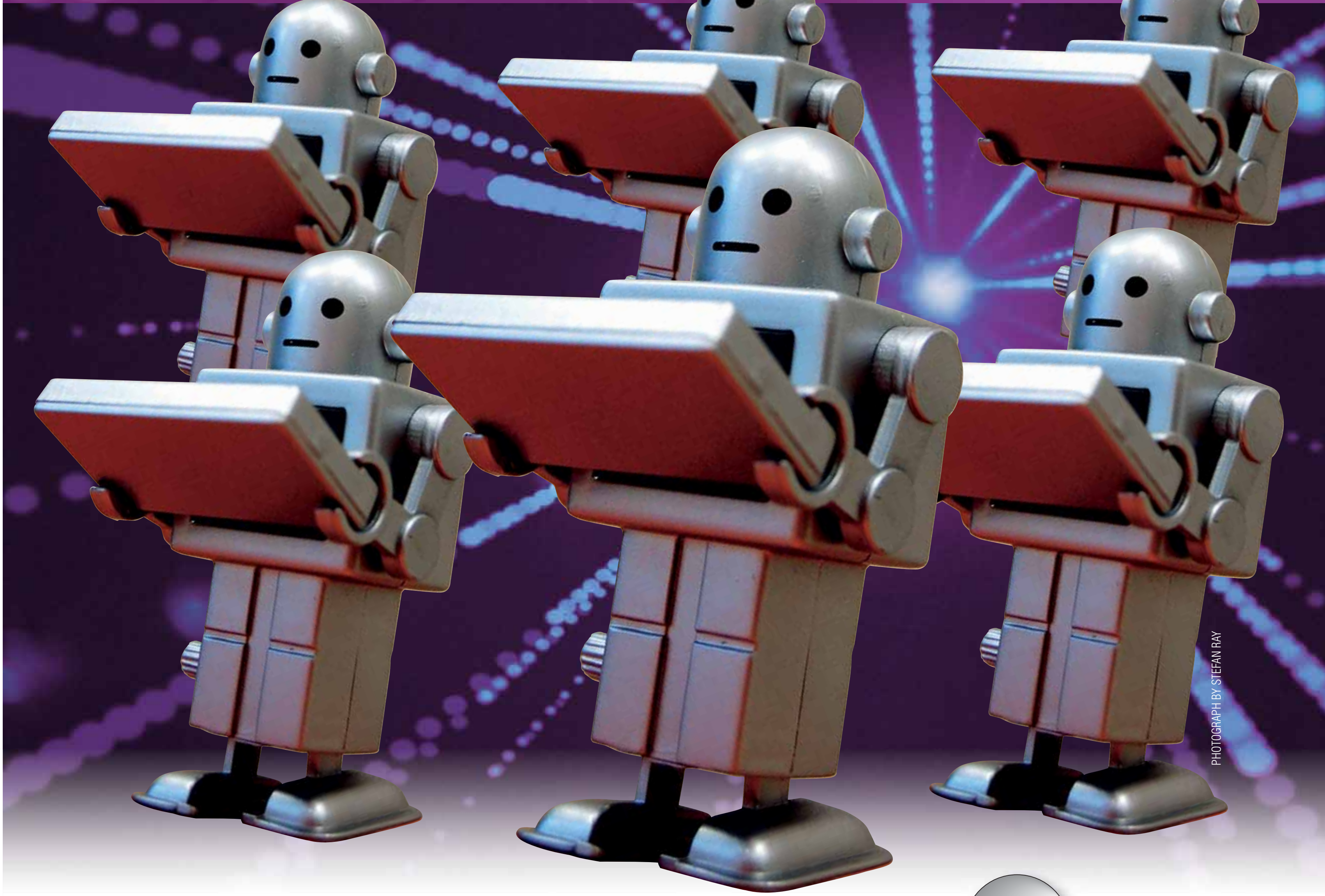
Вторую карточку, однако, мы не сможем распечатать через PostScript. *Inkscape* умеет сохранять PostScript с градиентами, но только если они не содержат прозрачности. Хотя в версии 0.44 есть экспорт в PDF с поддержкой прозрачности, для печати этот формат почти бесполезен. Поэтому самый надежный способ для второй карточки – экспорт в растровый формат высокого разрешения (A). Полученный PNG-файл можно конвертировать в CMYK TIFF и послать в принт-бюро или же распечатать на настольном принтере с помощью растрового редактора вроде *Gimp*. LXF

ПОДСКАЗКИ

- Чтобы *Inkscape* увидел новый шрифт, просто инсталлируйте этот шрифт в вашей ОС и перезапустите *Inkscape*. Новый шрифт появится в диалоговом окне Text And Font (Текст и Шрифт).
- Большинство текстовых объектов можно улучшить настройкой трекинга (стандартного расстояния между буквами) и кернинга (расстояния между отдельными парами букв). В инструменте Текст нажимайте Alt+< и Alt+> для регулировки трекинга в выделении и Alt+стрелки для изменения кернинга под текстовым курсором.
- Если вы хотите конвертировать растровый выходной файл в формат CMYK TIFF для печати, это можно сделать с помощью только свободных программ. Сначала конвертируйте PNG в RGB TIFF (с помощью *Gimp* или *ImageMagick*), затем воспользуйтесь командой tifficc из библиотеки LittleCMS (www.littlecms.com) для конвертации его в CMYK TIFF. Для этого преобразования вам понадобится ICC-профиль вашего выходного устройства.

ЧЕРЕЗ МЕСЯЦ

Продолжая оттачивать мастерство, мы создадим настоящий анимированный баннер в *Inkscape*. Не пропустите!


PHOTOGRAPH BY STEFAN RAY

АВТОМАТИЗАЦИЯ OPENOFFICE.ORG СЕРИЯ: «СЦЕНАРИИ НА BASIC»

НА ДИСКЕ • Код из учебника

OOo Basic Работа с базой данных

ЧАСТЬ 3 Очередной шанс поумнеть при помощи **Марка Бэйна** – на этот раз его макросы и советы по работе с базами данных помогут вам запускать запросы, создавать отчеты в OpenOffice.org и вести учет ваших книг и

МЕСЯЦ НАЗАД



Мы занимались изучением OOo Basic в табличном процессоре ***Calc***.

До чего ж я люблю *OpenOffice.org* – особенно когда использую его вместе с OOo Basic. Не только потому, что он высвобождает меня из клещей ProprietarySoft, Inc – больше потому, что он действительно хорош. В предыдущих выпусках мы увидели, как легко можно манипулировать текстовыми документами и таблицами с помощью OOo Basic. На сей раз –посмотрим, как извлекать информацию из базы данных.

Главное, надо быть как можно ленивее. Представьте, например, что вы хотите подготовить счет для ужасно популярного Linux-журнала, в который вы пописываете. Зачем терять время, перепечатывая то, что у вас уже хранится? Это руководство даст вам инструменты, пресекающие лишнюю трату времени, а заодно, естественно, позволит насладиться исследованием Unix.

Ингредиент №1: сервер базы данных

Начинать – так с начала. Раз уж это руководство по макросам для извлечения информации из базы данных, вам понадобится база данных. Однако я не намерен рассказывать о ее установке: это выходит за рамки руководства. Конечно, если бы вы объявили, что базы данных у вас нет и вы даже не знаете, с чего начать, то я ответил бы: «Без паники, это легко». Затем я предположил бы, что вам нужен сервер баз данных – так ведь можно использовать любой старый компьютер, подсоединить его к сети и затем установить Debian (если у вас нет второго компьютера, запустите сервер на своей машине). Вы сами можете создать минимальный загрузочный диск с **www.debian.org**, вставить в привод, перезагрузиться и следовать инструкциям. Об установке дополнительного программного обеспечения (рабочего стола, файл-сервера, web-сервера и т.д.) беспокоиться нечего: достаточно необходимого минимума.

Тут я велел бы вам превратить ваш компьютер в сервер баз данных с помощью команды apt-get install mysql-server, а затем отредактировать файл **/etc/mysql/my.cnf**, закомментировав строку bind-address = 127.0.0.1 (чтоб она выглядела как #bind-address = 127.0.0.1). Это позволит подключаться к серверу с любого компьютера вашей сети.

Вам, небось, захотелось бы создать и базу данных, и пользователя для доступа к ней. Тогда бы я посоветовал сделать следующее:

```
mysql -uroot mysql
set password for 'root'@'localhost' = password('put your own password here');
create database accounts;
grant all privileges on accounts.* to 'your user'@'%'
```

```
identified by 'your user password';
exit;
```

Наконец, я предложил бы задать вашему новому серверу статический IP-адрес, отредактировав файл **/etc/network/interfaces** так, чтобы конец файла был похож на следующее:

```
#iface eth0 inet dhcp
iface eth0 inet static
      address 192.168.1.3
      netmask 255.255.255.0
      gateway 192.168.1.1
```

В этом пункте я бы отметил, что вам пора перегрузиться и зайти на компьютер, где у вас стоит *OpenOffice.org*.

Но так как наше руководство исключительно про OOo Basic, а не про создание баз данных, всего этого я делать не буду.

Доступ к базе данных

OpenOffice.org пока не запускайте. Чтобы облегчить себе жизнь (для того и придуманы макросы), воспользуемся UnixODBC, это API для доступа к источникам данных, который избавит нас от трудностей создания соединений к серверу и базам данных – протоколы, посылка сигналов и все такое прочее. Самое сложное, что предстоит сделать – это установить UnixODBC и его библиотеки MySQL на машину, где вы будете использовать OOo. На Debian это делается всего-навсего через

```
apt-get install unixodbc
apt-get install libmyodbc
```

Очевидно, если у вас другой дистрибутив, то придётся проверить для него процесс установки – взгляните на UnixODBC на странице **www.unixODBC.org**. Когда вы установите UnixODBC, понадобятся еще две вещи. Первое – отредактировать **/etc/hosts** так, чтобы он включал ссылку на сервер вашей базы данных, то есть 192.168.1.3 acamas. Второе – отредактировать **/etc/odbc.ini**, чтобы он включал примерно следующее:

```
(accounts)
Description     = MySQL db test
Driver          = MySQL
Server          = acamas
Database        = accounts
Port            = 3306
```

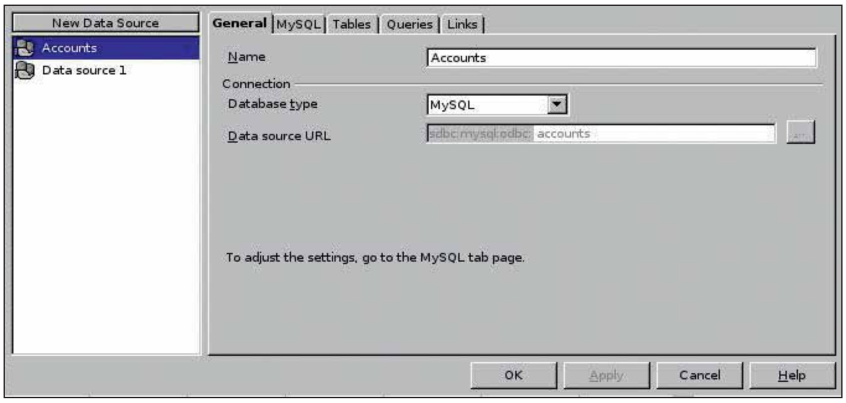
Теперь – глубокий вдох, сосчитать до пяти, медленный выдох, и готово дело: нет больше командных строк.

Разборки с базой данных

Откройте OpenOffice.org. Тип документа безразличен; пусть, например, это будет документ Writer. В меню Tools [Сервис, иногда нужный вам пункт оказывается в меню View (Вид), – прим.ред.] среди подменю имеется одно под названием Data Sources (Источники данных). Кликните на нем, и вы увидите форму Data Source Administration (Управление источникам данных).

С ней все просто: нажмите New Data Source (Новый источник данных) и установите тип базы MySQL на вкладке General (Общие). Затем перейдите на вкладку MySQL, добавьте имя базы данных в Data Source URL (Адрес источника данных) и введите имя пользователя (не забудьте создать пустую базу данных и пользователя, прежде чем получать к ней доступ из OOo). Далее нажмите на закладку Tables (Таблицы). Там ничего не будет (потому что никаких таблиц вы еще не создали). Угадайте, что мы теперь сделаем? Правильно, рванём в пивбар, с вас причитается. Нет? Ладно, оставим это на потом: время создавать данные.

Если вы любитель командной строки (должен сознаться, что я из них), вернитесь на сервер с базой данных, зайдите в базу и создайте таблицы. Не забывайте, что вы можете подключиться прямо с текущей машины.



Надо заранее знать, к какой базе данных подключаться из *OpenOffice.org*.

```
bainm@hector:~/ooobasic3$ mysql -hacamas -ubainm -pmypassword accounts
mysql > create database accounts;
create table accounts.customer (id int auto_increment,
surname varchar(50), firstname varchar(50),
address1 varchar(50), address2 varchar(50), city varchar(50), county varchar(50),
country varchar(50), postcode varchar(50),primary key (id));
create table accounts.invoice (id int auto_increment,customer_id int,
sent_date date,paid_date date,primary key (id));
create table accounts.item (id int auto_increment,customer_id int,
invoice_id int,title varchar(50),details varchar(255),value double,
primary key (id));
insert into  accounts.customer
(surname,firstname,address1, address2,city,county,country,postcode)
values
('Smith','John','The Big House','1 The Street','Thistown','Thisshire','UK','TH 1 1HT');
insert into  accounts.customer
(surname,firstname,address1, address2,city,county,country,postcode)
values
('Jones','Mary','Building A','Industrial Est.','Hereton','Herehire','UK','HE1 1EH');
insert into accounts.item (customer_id,title,value) values (1,'A fine piece of work',500);
insert into accounts.item (customer_id,title,value) values (1,'A great job',500);
insert into accounts.item (customer_id,title,value) values (2,'Day 1',1500);
insert into accounts.item (customer_id,title,value) values (2,'Day 2',1600);
```

Кого это в восторг не приводит, пусть возьмёт форму Data Source Administration (Управление источникам данных), перейдёт на вкладку Tables (Таблицы) и нажмёт на кнопку New Table Design (Создать новую таблицу). Можете воспользоваться формой Table Design (Создание таблицы), с её помощью таблицы создавать легко.

Работа с новыми таблицами

Мы извели довольно много времени на разборки с базой данных – без её правильной работы не обойтись, а все остальное само встанет на свои места. Теперь можно заняться нашим первым макросом для работы с базой данных. Если вы следовали руководству прошлого выпуска, то привыкли к функции OOo CreateUnoService (вы ведь практиковались, не так ли?). Мы снова собираемся использовать ее здесь, на сей раз для доступа к RowSet. Это имя OOo для набора записей, оно позволяет выполнять запросы к базе данных и получать от нее информацию.

```
RowSet = createUnoService("com.sun.star.sdb.RowSet")
```

Осталось только сказать RowSet о базе данных, к которой вы хотите подключиться (то есть к той, что вы установили в Data Source Administration (Управление источникам данных)): сообщите имя пользователя, пароль и запрос, который хотите выполнить. RowSet получит результат запроса и представит его вам.

Поэтому вы, видимо, захотите сделать следующее:

```
sub main
  sql1
end sub
Sub sql1
```

>>

«СВОИ Ж БЕЗУМСТВА ИХ И ПОГУБИЛИ»

Удивлены выбором имен хостов? Они из Илиады Гомера. При всей моей любви к Властелину колец (источник большинства имен хостов) я нахожу удивительным, что история из бронзового века так схожа с сегодняшним днем и что человечество ничуть не изменилось за этот долгий срок.

```
  Dim RowSet
  RowSet = createUnoService("com.sun.star.sdb.RowSet")
  RowSet.DataSourceName = "Accounts"
  RowSet.User="bainm"
  RowSet.Password = "password"
  RowSet.Command = "SELECT count(*) c FROM item"
  RowSet.execute()
  RowSet.next()
  MsgBox "There are " + rowSet.getString(1) + " items"
End Sub
```

Отлично, теперь рассмотрим следующий пример:

```
Dim RowSet
Sub Main
      connectToDatabase ("Accounts", "bainm", "kawasaki")
      sql1
End Sub
Sub connectToDatabase(database as string, username as string, password
as string)
  RowSet = createUnoService("com.sun.star.sdb.RowSet")
  RowSet.DataSourceName = database
  RowSet.User = username
  RowSet.Password = password
End Sub
Sub updateRowSet(sql as string)
  RowSet.Command = sql
  RowSet.execute()
End Sub
Sub sql1
  updateRowSet("SELECT count(*) c FROM item")
  RowSet.next()
  MsgBox "There are " + rowSet.getString(1) + " items"
End Sub
```

Теперь понятно, как легко расширить функциональность макроса. Взгляните:

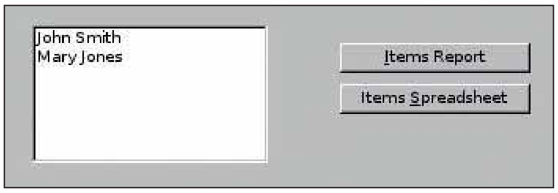
```
Sub sql2
            updateRowSet("SELECT id, surname, firstname  FROM
customer")
            while RowSet.Next()
                        MsgBox "Customer No. " + rowSet.getString(1) +
" " + rowSet.getString(2) + _
                        " " + rowSet.getString(3)
            wend
End Sub
```

Написание отчетов

Мы увидели, что с помощью макроса легко получить доступ к базе данных и отобразить результаты. Но пока не увидели ничего такого, чего нельзя сделать столь же легко из командой строки. Вспомните LXF80: там мы осуществляли запись напрямую в документы *OOo Writer*. Видимо, неглупо будет аналогично поступить с информацией из нашей базы данных.

«Я УВЕРЕН: ВЫ ПОЙМЕТЕ, ЧТО ВСЁ ЭТО ОЧЕНЬ ПРОСТО...»

Замечательно то, что мы можем делать впечатляющие вещи добавкой всего нескольких строк кода. Мы уже разбирали процедуру loadNewFile (мы познакомились с ней в LXF80 и модифицировали в LXF81) для создания нового документа *Writer*, и у нас есть процедура add_paragraph для записи в документ (не пугайтесь, весь нужный код содержится на прилагаемом диске). Надо только добавить процедуры для создания отчетов по информации в базе данных. Вот простой способ создания документа, содержащего список всех покупателей в базе данных Accounts:



Симпатичное окошечко управляет вашими отчетами.

```
Dim RowSet
Sub Main
            connectToDatabase ("Accounts", "bainm", "kawasaki")
            loadNewFile
            createCustomerReport
End Sub
Sub createCustomerReport
            updateRowSet("SELECT id, surname, firstname  FROM
customer")
            while RowSet.Next()
                        add_paragraph("Customer No. " + _
                        rowSet.getString(1) + " " + rowSet.getString(2) +
" " + rowSet.getString(3))
            wend
End Sub
```

Вот и вся любовь. Процесс прост: посылаете запрос в базу данных, а затем отображаете результат в документ. Конец истории? Вообще-то не совсем. В LXF80 мы обнаружили: никто не любит менять функцию Main под создание каждого нового отчета – ну разве что мазохисты. И снова, ключевым моментом является создание диалогового окна для управления требуемыми работами.

Вам уже не понадобится вручную набирать содержимое элементов в виде списка. Нет, на этот раз вы загрузите их прямо из базы данных. Представим, что вы добавили элемент list box и назвали его lstCustomers в диалоговом окне dlgAccounts. Чем его загружать? Вы меня опередили: можно послать запрос в базу данных на получение списка покупателей:

```
            updateRowSet("SELECT surname, firstname  FROM
customer")
```

Теперь в цикле переберите набор записей и загрузите их в элемент list box:

```
            lstCustomers.AddItem(rowSet.getString(2) + " " + rowSet.
getString(1), i)
```

Просмотрите процедуру loadlstCustomers, расположенную на нашем диске, чтобы разобраться, как она работает.

Новый элемент list box пригодится как фильтр для создания произвольных отчетов. Допустим, вам захотелось увидеть все предметы, купленные определенным покупателем – ну так используйте свойство selectedItem из list box и получите выбранный текст, а затем примените его для создания SQL-запроса:

```
sql = " select title,value from customer, item " + _
            " where cutomer.id = item.customer_id " + _
            " and concat(customer.firstname,concat(' ',customer.surname))
= '" + _
            lstCustomers.selectedItem +"'"
```

Ещё лучше встроить SQL в функцию. Зачем? Таким образом вы сможете использовать запрос в любой процедуре без необходимости переписывания кода. Теперь добавьте в окно кнопку, ассоциируйте с ней процедуру и начинайте пользоваться. Для начала сделайте процедуру, выводящую на экран окно с вашим построенным SQL-запросом. Теперь используйте SQL для получения нового набора записей и запишите итог в документ *Writer*. cmdItemReport с нашего диска покажет вам это в действии.

Я уверен: вы поймете, как всё это просто (запомните хорошенько: это просто), и автоматизация извлечения информации из базы данных в документ *OOo Writer* тоже проста. Вас, наверное, не удивит, что данные можно передавать и в таблицу *Calc*. Взаимодействие с базой дан-

ных происходит таким же образом. Единственное отличие – вы должны писать в отдельные ячейки, а не в абзацы, а это даёт даже больше гибкости в отображении вашей информации.

А теперь я вас покидаю – придумайте сами, что теперь делать: всё необходимое мы обсудили в LXF80, LXF81 и в этом выпуске. И если вы всё ещё в тупике, то загляните в раздел «Журнал» на диске – готовые программы уже ждут вас не дождутся.

Медиа-библиотека

На закуску рассмотрим простое приложение – оно поможет вам хранить и просматривать список всех ваших CD- и DVD-дисков, пластинок или книг.

Начните с создания таблиц в вашей базе данных. Вам придется задаться вопросом: работать ли с отдельной базой данных для каждого проекта или поместить все таблицы в одну базу? Я бы порекомендовал первое – так проще управлять информацией. Однако, выбрав этот метод, не забудьте добавить запись о новой базе в **/etc/odbc.ini** и добавьте ее как новый источник данных в *OpenOffice.org*. Понадобится также подсказать макросу, чтобы он использовал новую базу данных – поменяв connectToDatabase (“Accounts”, “bainm”, “kawasaki”) на connectToDatabase (“library”, “bainm”, “kawasaki”).

Далее: не пытайтесь вбить всё в одну таблицу – получите только проблемы. Какие именно? Что ж, давайте рассмотрим простой пример – поле, содержащее имя. Вы-то знаете, что Б Гейтс, Уильям Гейтс и Властелин Зла означают одно и то же лицо, но ваш компьютер не знает, и это затруднит процесс создания запроса. Взгляните на таблицу:

Таблица: item

Title (Название)	Author (Автор)
Колыбель для кошки	Курт Воннегут
Бойня номер 5	К Воннегут

Взамен можно использовать две таблицы – одна с описанием предметов, другая с авторами:

Таблица: item

Title (Название)	Author (Автор)
Табакерка Багомбо	1
Сирены Титана	1

Таблица: author

ID	Name (Имя)
1	Курт Воннегут-младший

Таким образом, вместо запоминания всевозможных написаний имени автора вы обойдётесь его идентификационным номером. Аналогично, вам не надо хранить слова cd, lp, book в таблице, содержащей заголовок. Вместо этого можно использовать что-то вроде:

Таблица: item

Title	Media ID (Тип носителя)
Бомба для мозгов	2
Дзен и искусство ухода за мотоциклом	1

Таблица: media

ID	Type
1	Book
2	CD

Теперь с помощью SQL-запроса вы можете получить из базы данных полезную информацию:

```
select item.title, author.name, media.type
from item, author, media
where item.author_id = author.id
and item.media_id = media.id;
```

Используйте этот SQL-запрос в процедуре заполнения таблицы результатом запроса – изучите showFullLibrary на диске, чтобы понять, как это работает (там же вы найдете SQL-запрос для создания базы данных и пример файла **/etc/odbc.ini**). Внимательно посмотрев на этот макрос, вы обнаружите, что в нем не содержится жестко заданного числа столбцов, когда осуществляется запись в документ; вместо этого для создания цикла используется свойство RowSet.Columns.Count. И что? А вот что: неважно, если вы измените число записей, получаемых в запросе – макрос автоматически вставит правильное число столбцов в таблицу.

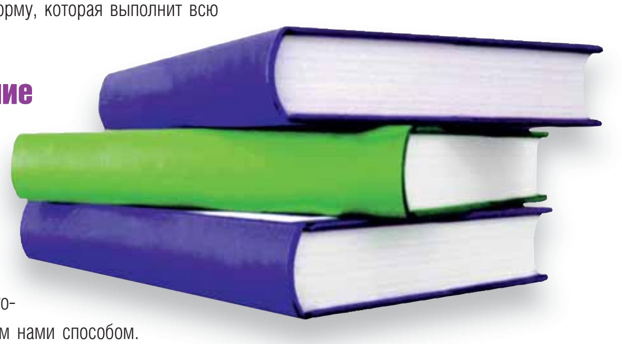
Фильтрация данных

Так и слышу ваш крик: «Да не хочу я видеть все, что содержится в базе данных! Мне надо смотреть только CD-диски, или только книги, или только работы одного художника». Что ж, легко – если вы создадите новую форму, то можете добавить на нее несколько элементов list box и заполнить их из таблиц author и media (так же, как мы сделали в примере с покупателями). Элементы list box можно использовать как фильтры для построения запроса. На диске, showFilteredLibrary показывает, как использовать опциональный ввод для построения такого фильтра и последующего отображения результата в таблицу.

Чтобы добавить в базу новые предметы, авторов или типы носителей, вам пригодится выражение insert, например:

```
insert into  library.author (name) values ('Hawkwind');
insert into  library.item (title,author_id,media_id) values ('The Ambient Anarchists',4,1);
```

Можно это сделать и из командной строки, но приобретённые знания помогут вам создать форму, которая выполнит всю работу за вас.



Домашнее задание

Задание на месяц (и никаких экивоков типа «если у вас будет время, то...» – это нужно не мне, а вам): проанализируйте свои каждодневные задачи и выберите те, которые можно автоматизировать рассмотренным нами способом. Не ради увеличения производительности и тому подобной ерунды, а исключительно из лени. Рекомендую также взглянуть на запросы update – почему бы не записывать данные в базу так же, как и читать их из нее? LXF

ЧЕРЕЗ МЕСЯЦ

Советы по OOo Basic (и может быть, я расскажу вам про эти запросы update).



РАЗРАБОТКА СОБСТВЕННОЙ 3D-ИГРЫ

Ogre Создаем ландшафт для 3D-игры

ЧАСТЬ 1 **Пол Хадсон** начинает это долгожданное руководство с описания ландшафта.

НА ДИСКЕ

- Ogre 3D 1.2
- DevIL 1.5
- zzlib 0.12
- Драйверы для видеокарт
- Исходный код примеров статьи

Что приятно в написании статей для Linux Format так это необходимость расширять собственный кругозор для самой возможности создания и описания новых проектов для ваших тренировок. Можете себе представить, к примеру, как я был озадачен, когда меня попросили заняться серией статей о программировании 3D-игр. Не то что я не люблю писать о Linux: на самом деле, очень люблю. Но больше я, конечно, люблю играть в игры, потому как игры – это весело по определению. И вдруг оказалось, что веселее, чем играть в игры, только одно (по крайней мере, на компьютерах): их создавать! Обладание неограниченной исполнительной властью над жизнью проекта, несущего счастье миллионам* .

Поэтому мы, в Лаборатории LXF, составили базовое руководство по проектированию, разработке и выпуску вашей собственной 3D-игры. У вас, вероятно, уже руки чешутся поработать, но не спешите – в данной статье я сначала поведаю об игре вообще и о программах с библиотеками – необходимом инструментарии для ее разработки, а уж потом напущу вас на решение библейски-первозданной задачи: наколдовать для игры небо и землю.

Барьер номер один

Прежде чем браться за написание кода – или даже за установку нужного для кодирования программного обеспечения, надо сообразить, что именно вы хотите написать (дико извиняюсь перед программистами-экстремалами). Поэтому я задам особенности предполагаемой игры: что она будет делать, как выглядеть, как в нее играть, и так далее.

- Это будет стрелялка от первого лица (First-Person Shooter, FPS). Я знаю, существуют тысячи FPS, а все потому, что их сравнительно легко порождать, они дают разгуляться фантазии и нуждаются в безумно красивой графике. Я не намерен ничего придумывать за вас, но надеюсь создать игру, простую в разработке и классную на вид!
- Это будет однопользовательская игра с неким подобием искусственного интеллекта, чтобы добавить сложности.
- Действие будет происходить на обширной открытой местности, игроки будут ее исследовать. Для разнообразия, они смогут передвигаться на транспорте, а может, и входить в помещения. Так игра будет больше похожа на *Quake*.
- Игра будет написана на C++, по принципу «лишь бы работало». За совершенством гнаться не станем. Место в журнале ограничено, и я лучше покажу вам 5 строк нормально работающего кода, чем 50 строк идеального.
- Игра будет кросс-платформенной. Мы будем разрабатывать ее в SUSE 10.1, но она должна работать и на любом другом дистрибутиве Linux. Будет здорово, если игра также будет работать в Windows и OS X, но это не главная наша цель.
- Игра будет выпущена под лицензией GPL.

*Миллионам? Да, люблю метить высоко. Поэтому ничего не замечаю, пока не грохнусь.

Напоминаю, что это руководство по программированию 3D-игр, а не по C++, математике или *Blender*. Поэтому я сфокусируюсь на 3D-графике, прихватывая готовые модели и элементы игры везде где только можно. Если вы знаете *Blender* – прекрасно: сможете создать собственное творение. Если нет, не переживайте: все, что вам нужно – способность программировать.

Наша цель состоит в создании законченной игры. Для этого мы воспользуемся графическим движком Ogre 3D и SDL для аудио и прочих нужд. Однако потребуется еще заполнить немало пробелов, оставленных SDL и Ogre.

Список требований

Теперь вы знаете, что мы хотим запрограммировать, поэтому необходимо привести в боевую готовность вашу систему. Для программирования потребуется установить следующее ПО (заметим, что здесь указаны лишь минимально допустимые версии; более новые версии будут предпочтительнее):

- *Automake 1.6*
- *Autoconf 2.5*
- *make 3.8*
- *libtool 1.4*
- *pkg-config 0.17.2*
- *GCC 3.4*
- *g++ 3.4*
- *cpp 3.4*

Приведенные названия должны совпадать с теми, что вы найдете в менеджере пакетов вашего дистрибутива. Исключение может составить *pkg-config*, иногда называемый **pkgconfig**.

Так как мы собираемся работать с играми, понадобятся дополнительные пакеты, а именно:

- *SDL 1.2.9*
- *SDL_Mixer 1.2.6*
- *Mesa*
- *FreeType2 2.1*
- *libpng*
- *libmng*
- *libtiff*
- *libjpeg*

Как и в первом случае, все вышеозначенные библиотеки можно найти в менеджере пакетов. Проверьте, что вы устанавливаете как саму библиотеку, так и версию для разработчика, иначе вы сможете только запускать игру, но не разрабатывать свою собственную. Например, помимо пакета SDL существует пакет **SDL-devel** (или **SDL-dev**), который также надо установить.

Теперь нам надо установить действительно особые программы, предназначенные для разработки игры. Этот список гораздо короче – и, возможно, не все они найдутся в вашем менеджере пакетов. Нам понадобятся следующие:

- *zziplib 0.12*
- *DevIL 1.5*
- *Ogre 3D 1.2*
- Приличный драйвер видеокарты

Мы разместили все эти пакеты на диске (включая последние драйвера Nvidia и ATI для Linux) и расскажем, как их установить. Начнем с драйвера видеокарты. У меня MSI Nvidia GeForce 7900, поэтому весь код, описанный здесь, будет работать с картами Nvidia.

Убойный графический драйвер

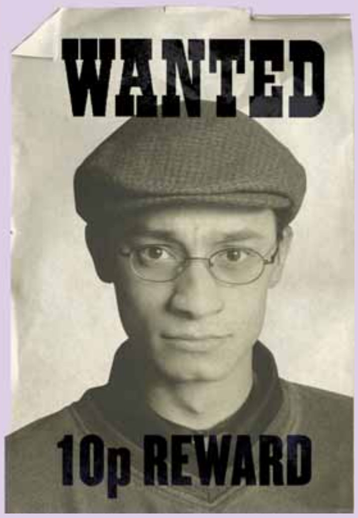
Чтобы установить драйвер Nvidia, нажмите Ctrl+Alt+F1 – попадете в терминал. Переключитесь в суперпользователя и наберите init 3, чтобы завершить работу X. Далее скопируйте драйвер с нашего диска в свой домашний каталог и запустите **sh ./NVIDIA-Linux-x86-1.0-8756**. Запустится программа установки, с текстовым интерфейсом, и вас попросят принять лицензионное соглашение. Если драйвер у вас уже был, вам надо нажать Yes, чтобы его удалить.

ЗНАКОМЬТЕСЬ С ВАШИМ ГЕРОЕМ: ЧЕД-В-БЕГАХ

Хотя большинство стрелялок от первого лица проходят по формуле «целься, стреляй, беги, опять стреляй», они все-таки пытаются обставиться хоть каким-то сюжетом. И вот наш сюжет, в виде краткого представления персонажа:

Десять лет назад лихой спецназовец был приговорен судом к тюремному заключению за преступление, которое не совершал, но быстренько сбежал из-за колючей проволоки к нелегалам Сан-Франциско. Сейчас, будучи все еще в розыске, он работает солдатом удачи. Если у вас проблемы, и помочь вам некому, сумейте отыскать и нанять... Чеда!

Наш герой Чед приговорен к повешению и находится в бегах. Игра будет патетически называться «Висельник Чед», хотя в идеале наш герой избежит казни, изувечив всех на своем пути. Он будет бороться против плохих парней (в основном против тех, что для нас заготовили разработчики Ogre), преодолевая всяческие трудности, для чего потребуется настоящий героизм (но главным образом пальба). Кроме того, он задействует арсенал тяжелого боевого оружия, позволив нам поупражняться в спецэффектах. Заинтересовались? Еще бы!



Может оказаться, что для вашего ядра не найдется скомпилированного интерфейса, поэтому нажмите Yes, чтобы поискать драйвер на сайте Nvidia. Если его там нет, снова нажмите OK и скомпилируйте собственный драйвер. Наконец, будет предложено отредактировать за нас файл настроек X – соглашайтесь, да не забудьте потом проверить файл **/etc/X11/xorg.conf** и убедиться, что используется драйвер 'nvidia' (а не 'nv' или, того хуже, 'vesa'). По завершении установки наберите init 5 – вернетесь обратно в X. Откройте терминал и запустите glxgears. Секунд через пять вы увидите результаты тестирования – если результат меньше 1000 кадров в секунду, это сильно повредит нашей игре.

Для более продвинутой работы с графикой можете установить библиотеку *Cg* (здесь некогда про нее рассказывать, но вреда от нее всяко не будет). 'Cg' – сокращение от 'C for Graphics', а цель библиотеки – облегчить создание продвинутых визуальных эффектов. Вы можете скачать инструментарий *Cg* со страницы Nvidia для разработчиков по адресу **http://developer.nvidia.com/page/cg_main.html**. Процесс ее установки лишен особого полета: скопировать в домашний каталог в подкаталог **Cg**, извлечь с помощью **tar xvfz Cg-1.4.1_x86.tar.gz**, переключиться в суперпользователя и запустить **cp -R usr /**.

Установка *zziplib* – тоже всего-навсего извлечение и запуск **./configure**, **make** и **make install**. Если кому интересно, *zziplib* предоставляет быстрый и удобный способ манипулирования zip-файлами, а именно в них *Ogre* хранит множество своих ресурсов.

Далее идет *DevIL* – мультиплатформенная библиотека для работы с изображениями, используемая *Ogre*. Она находится на диске к журналу, так что скопируйте ее в ваш домашний каталог и запустите

```
tar xvfz DevIL-1.6.8-RC1-src.tar.gz
cd DevIL-1.6.8-RC1
./configure
make
su
make install
exit
```

Теперь осталось главное событие: *Ogre*. Скопируйте его в домашний каталог с диска и наберите:

```
tar xvfj ogre-linux_osx-v1-2-0.tar.bz2
cd ogrenew
./bootstrap
./configure (или
./configure --with-platform=GLX если используется Nvidia)
```

>>

<<

```
make
su
make install
ldconfig
exit
```

Список требований выполнен – можно начинать делать игру, ура!

Время кодировать

Настал долгожданный момент первой «вылазки» в C++ ! Наш первый урок – не простое введение или руководство по установке: мы собираемся написать сердцевину нашего движка. Цель урока – создание некого ландшафта, по которому можно перемещаться с помощью мыши, и неба над ним. Большая часть кода этого урока состоит из основных начальных установок *Ogre*: инициализации движка, обработки ввода и т.д. В следующих выпусках уже добавятся красивости – например, управление с клавиатуры, туман и свет, анимация и спецэффекты; а пока займемся черновой работой.

Вооружившись этими данными, откройте текстовый редактор и настучите следующий код в **chad.h**, основной заголовочный файл нашего игрового класса:

```
#include "Ogre.h"
#include "SDL/SDL.h"
#include "SDL/SDL_mixer.h"
using namespace Ogre;
#include "chadframelistener.h"
class CChadGame {
public:
CChadGame();
~CChadGame();
bool loadConfig();
void initialise();
int run();
void createScene();
Root* m_Ogre;
EventProcessor* m_EventProcessor;
CChadFrameListener* m_FrameListener;
SceneManager* m_SceneMgr;
Camera* m_Camera;
Viewport* m_Viewport;
};
```

Первые три выражения **#include** включают наши стандартные библиотеки, хотя SDL мы пока трогать не будем. Четвертый **#include** предназначен для обработчика кадра –скоро мы к нему вернемся. Далее идет главный класс нашей игры, ответственный за большую часть ее организации. Не буду выпендриваться с объектно-ориентированным программированием, хотя как раз организации объектный подход и способствует – как я уже сказал, нам важно не изящество кода, а удобство игры; итак, не удивляйтесь, что я браво игнорирую инкапсуляцию объектов ради экономии места!

Класс **CChadGame** содержит обработчик событий и обработчик кадра, наш менеджер сцены (отслеживающий все объекты), камеру (позицию игрока) и область просмотра (то, что мы видим на экране). Позже в нем появится информация об игроке, очках, картах и многое другое, но сейчас все предельно просто.

Большая часть работы совершается обработчиком кадра и менеджером сцены, во многом благодаря *Ogre*. Задача нашего класса **CChadGame** состоит в том, чтобы установить каждую сцену и обрабатывать любые изменения, например, передвижение игроков. Однако менеджер сцены параллельно будет отслеживать все объекты в нашей игре и обеспечивать их корректную прорисовку (с применением оптимизации). Нашу заботу о вводе данных от пользователя любезно берет на себя обработчик событий *Ogre*. Затем эта информация посылается в обработчик кадра, который переваривает изменения.

Реализация класса **CChadGame** находится в файле **chad.cpp** на нашем диске, но он слишком велик, чтобы напечатать его здесь, поэтому отразим только основные моменты: конструктор (**CChadGame()**) и функцию **initialise()**.

```
CChadGame::CChadGame() {
m_Ogre = new Root;
this->loadConfig();
if(!m_Ogre->showConfigDialog()) return;
m_Ogre->initialise(true, "Hanging Chad");
this->initialise();
this->createScene();
}
```

Root – базовый класс *Ogre* и родитель остальных наших объектов из *Ogre*. Именно **Root** отвечает за прорисовку и передачу обратных вызовов к обработчику кадров. Именно объект **Root** показывает диалог конфигурации пользователю (через вызов **m_Ogre->showConfigDialog()**), а также вежливо завершает игру, когда мы сигнализируем, что с нас хватит. Первое, что мы создаем, и последнее, что удаляем – объект **Root**.

> «ИСПОЛЬЗУЙТЕ МЫШЬ, ЧТОБЫ ОГЛЯДЕТЬСЯ ВОКРУГ – НАШЕ ТВОРЕНИЕ ДОВОЛЬНО МИЛО!»

В функции **CChadGame::initialise()** настраиваются менеджер сцены, камера, область просмотра, обработчик событий и обработчик кадра. В ней же происходит вызов функции **InitialiseAllResourcesGroups()**, загружающей все ресурсы *Ogre*. Я одолжил стандартные файлы конфигурации (и их загрузчик, **CChadGame::loadConfig()**) из *Ogre* SDK и немного их подправил, чтобы они работали в нашей игре. Загрузчик конфигурации только читает текст: разборки синтаксиса не происходит, пока не вызовется **initialiseAllResourcesGroups()**. Если вы забудете вызвать эту функцию, то наверняка столкнетесь с проблемами!

Делай, что я сказал...

Осталось написать обработчик кадра, в который будут посылаться и обрабатываться события от *Ogre*. Здесь интерес представляют три функции, одна из которых говорит почти сама за себя. Это конструктор, и он сохраняет ссылку на устройство ввода, камеру и менеджера сцены для последующего использования.

Другая интересная функция – **frameStarted()**, вызываемая перед тем, как *Ogre* начнет обсчитывать кадр. Она перехватывает ввод и проверяет необходимость ответа на ввод от пользователя. В данный момент это значить «если нажали Escape, то возвратить false». Это значение завершит цикл прорисовки *Ogre*, и произойдет выход из игры. Третья функция – **mouseMoved()**, дублированная также в **mouseDragged()**; она вызывается при любом перемещении мыши, когда нам необходимо повернуть камеру в соответствии с параметрами перемещения.

Оставьте пока функцию **createScene()** пустой – и ваша программа, наконец, готова к сборке. Однако вручную эту работу выполнять довольно тягостно, поскольку необходимо подключить библиотеки SDL и *Ogre*. Поэтому обратимся к **Makefile**, чтобы вы могли набрать make и наслаждаться, любуясь, как вкалывают за вас. Наберите следующий текст в файл **Makefile** – но очень аккуратно, потому что пробелы – это на самом деле символы табуляции!

```
DEFINES =
LIBS = Ogre
CXX = g++
CXXFLAGS = $(shell pkg-config --cflags $(LIBS))
$(DEFINES) -I/home/paul/Desktop/Ogrenew/Samples/
Common/include
LD = g++
LDFLAGS = $(shell pkg-config --libs $(LIBS)) -lSDL -lSDL_
mixer -lpthread
```

Читатели *PC Plus*, возможно, помнят Чеда, но сейчас он в бегах и наверняка переодет в неброскую одежду, чтобы не бросаться в глаза.





1) Ogre позаботился об окне системных настроек, оно создается вызовом ***showConfigDialog()*.**

```
all:
$(CXX) $(CXXFLAGS) $(LDFLAGS) -o chad chad.cpp
clean:
rm -f chad
```

Время поколдовать

Все самое нудное позади, ваша «игра» (скопированная с диска к журналу) должна скомпилироваться и запуститься – просто наберите **make**. Когда она запустится, возникнет окно настроек *Ogre* (см. *рис. 1*). Рядом с пунктом Select Render вы увидите Select One. Нажав на него, вы получите список вариантов, доступных в вашей системе, где почти наверняка фигурирует только OpenGL Rendering Subsystem. *Ogre* все равно, как отрисовывать изображения, поэтому под Windows ваша игра может использовать *DirectX*.

Окно настроек позволяет сделать многое, например, установить разрешение экрана или выбрать полноэкранный режим; и *Ogre* автоматически сохранит ваш выбор в **Ogre.cfg** для дальнейшего употребления.

Настроив графику, нажмите Accept, и ваша «игра» начнется. Я написал слово «игра» в кавычках, потому что на данном этапе это пустой экран – зрелище не шибко захватывающее, но по крайней мере видно, что код работает! И теперь можно применить возможности *Ogre*: создадим пейзаж с помощью всего одной строки кода. Не бойтесь, она много короче тысячи символов.

Вот она, просто вставьте ее в вызов **createScene()**:

```
m_SceneMgr->setWorldGeometry(«terrain.cfg»);
```

Перекомпилируйте и запустите ваше приложение. Вы увидите нечто похожее на *рис. 2* – холмистый ландшафт. Свет, правда, прилип к текстуре, но даже так все выглядит вполне мило – это благодаря файлу **terrain.cfg**, который загружает уже существующие текстуры из *Ogre* SDK. Каталог, из которого берутся текстуры, получается из каталога установки *Ogre* вашей машины, а загружаются они в файле **resource.cfg**, который вы можете настроить по своему усмотрению (или позволить другим это делать), причем для этого не придется переписывать ни строчки кода.

Если вы все еще не пришли в восторг, добавим немного неба. Небо предусмотрено в трех моделях: плоское, коробка или купол (в порядке возрастания степени реализма и, соответственно, потребления ресурсов). Плоское небо (по сути, многоугольник, висящий над игроком) почти не нагружает GPU, но выглядит убого – наш ландшафт не упрятывает линию горизонта, и сразу заметно, что оно именно плоское. Модель-коробка ликвидирует данный недостаток, помещая игрока внутрь куба и отображая небо на каждую его грань. Потребление ресурсов шестикратно возрастает, зато уж небо есть везде. У последней модели – купола – отсутствует «дно», а на «покрышку» для пущей иллюзии натянута текстура. Отсутствие дна может создать проблему, если игрок глянет вниз, но там уже есть ландшафт, так что все в порядке.

Код для невероятно сложной иллюзии выглядит так:

```
m_SceneMgr->setSkyDome(true,"Examples/CloudSky");
```

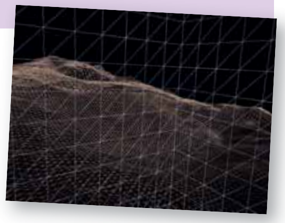
Да, да, это все! – теперь перекомпилируйте и запустите игру. Наслаждайтесь. Используйте мышь, чтобы оглядеться вокруг – вы увидите довольно милую картинку, как на *рис. 3*. Важнее всего, что наш небольшой продукт не так плох для начала! Теперь осталось добавить клавиши WASD для перемещения, немного тумана, пару источников света, злодеев и физи... Стоп, я, кажется, забежал впереди паровоза. Итак, вы получили краткое и несложное руководство для ознакомления с *Ogre* – и вас ожидает еще много интересного. LXF

ЗА КУЛИСАМИ...

Две строчки кода для генерации ландшафта набрать легко, но в ответ на них *Ogre* сгенерировал для нас аж 28000 треугольников. За крутыми строками упрятана напряженная работа – наша тестовая машина показала в среднем 600 кадров в секунду, а на менее мощном компьютере результат мог быть и хуже.

Здесь та же проблема, что и у интегрированных сред разработки, поддерживающих сворачивание кода. Вы видите строчку вызова и думаете: «Ага, тебя-то мне и надо», копируете ее и вставляете сразу в нескольких местах. А потом замечаете, что в свернутом коде на самом деле 1000 строк, и вы чуть ли не удвоили объем своей программы!

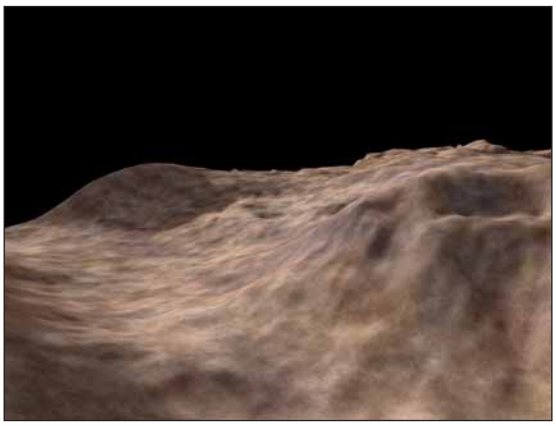
В *Ogre*, добавление одной строки может вылиться в тысячи, а то и миллионы новых треугольников, и всех их надо отобразить на экране. Поэтому будьте осторожны с поправками!



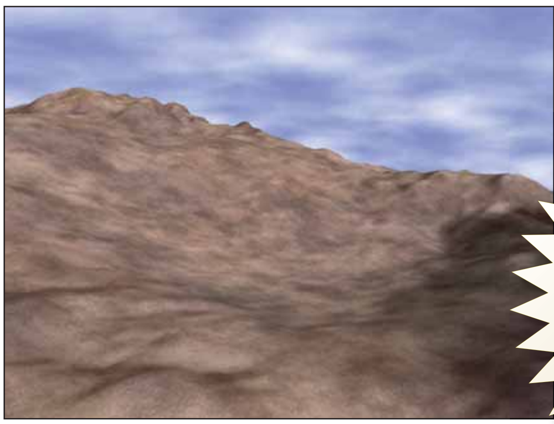
Наш маленький мир, в виде каркасной модели.

БЛАГОДАРНОСТЬ

Видеокарта Nvidia GeForce 7800, используемая для разработки этого руководства, была любезно предоставлена MSI . Спасибо, ребята!



2) Покатые холмы, прямо как в Уэльсе. Хотя не такие зеленые. И овец нет. Наверно, это террикон.



3) Таков конечный продукт данного этапа. Небо выглядит довольно убедительно.

ЧЕРЕЗ МЕСЯЦ

Мы позволим игроку пошататься по игровому полю и добавим обнаружение столкновений. Ниндзя-а-а!

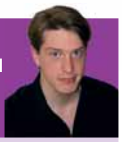


ПРОГРАММИРОВАНИЕ СЦЕНАРИЕВ

PHP Самое важное

В этой завершающей цикл статье **Пол Хадсон** дает нам несколько советов на посошок.

МЕСЯЦ НАЗАД



мы рассматривали расширение PHP для работы с SSH.

НА ДИСКЕ

• PDF-файлы со всей серией учебников PHP

Я начал писать первую статью этого цикла еще для *LXF30*. Не могу сказать, чтобы это было «будто вчера», поскольку это не так: автор письма месяца тогда получил коробку с Red Hat Linux 7.3, на диске к журналу находился релиз-кандидат Gnome 2.0, а в обзоре *KSpread* получил всего 5 баллов из 10 из-за того, что «в некоторых ситуациях в процессе тестирования *KSpread* рушился при попытке импортировать файлы, формат которых был объявлен как поддерживаемый». Ну хорошо, что касается *KSpread*, то это действительно было «будто вчера», но вот все остальное в том журнале кажется относящимся к давнему, давнему прошлому.

Конечно, главной темой той статьи было начало нового цикла про PHP, но тогда я понятия не имел, что эта серия окажется самой длинной за всю историю журнала. Это 53-я и последняя статья цикла, и в ней я хочу окинуть взглядом некоторые вещи, рассмотренные в предыдущих выпусках и дать несколько последних советов.

Дискография PHP

Если считать эти страницы, то у вас уже должно набраться больше 190 листов полезностей про PHP – этого почти достаточно, чтобы целиком заполнить два номера *LXF*! Мы рассмотрели огромное множество тем – генерацию картинок, базы данных, оптимизацию, сокеты, SNMP, базы данных, расширения, безопасность, графический интерфейс на основе *GTK*, опять базы данных, *curl*, проверку синтаксиса, *Gettext* и, конечно же, базы данных. Я надеюсь, что вы старательно занимали первое место в очереди за журналом каждый месяц, но если вы все-таки пропустили несколько выпусков (Эх!), то вам будет приятно узнать, что на диске вы найдете PDF-файлы со всеми до единого выпусками цикла, так что вы сможете прочитать все, что раньше пропустили.



Я хотел сделать читателей «специалистами по PHP 4.2» Увы, я не выполнил поставленной задачи – вы стали специалистами по PHP 5.1!

Глядя на последние 50 статей я полагаю, что если вам хочется углубиться в PHP, то я оставил для этого несколько заманчивых моментов. Пожалуй, моим любимым остается *LXF70*, в котором, пытаясь помочь людям победить в конкурсе *LXF*, я показал как разгадать загадку Монти Холла. Согласитесь, раньше вы не видели столько коз ни в одном из журналов. Ну и конечно никто не сможет забыть «священный» проект Interfict *(LXF57–62*), в котором мы на практике использовали множество приемов, изученных раньше, вроде баз данных и регулярных выражений. Но даже если у вас другие есть другие предпочтения, сегодня вы получите полный набор всех статей цикла. Наслаждайтесь!

Знаменитое последнее слово

Поскольку сейчас у меня есть последний шанс сказать вам что-то про PHP, я собираюсь дать вам несколько указаний, которым вам стоило бы следовать при программировании, и которые вам стоило бы учесть на будущее. Вот они:

1) Используйте *MySQL*. Это стандартный открытый сервер баз данных, он достаточно надежный и богатый возможностями, чтобы удовлетворить любого. Если вы используете что-то другое, потому что раньше оно было лучше, сейчас настало время пересмотреть ваше решение.

2) Используйте Ajax. Не стоит считать, что это «всего лишь еще одно модное слово», поскольку это не так: Ajax – это будущее web как минимум на ближайшее десятилетие. Я редко упоминал его тут, поскольку Ajax больше связан с JavaScript, но система LAMP (Linux, Apache, MySQL, PHP/Perl/Python) лучше других приспособлена к работе в качестве основы для Ajax-решения.

3) Используйте SimpleXML, когда вам нужно иметь дело с XML. Это одно из расширений PHP, которое действительно превосходно. Стоит вам начать разбираться с XPath – и вы уже вряд ли вернетесь назад. Если вы еще с не знакомы с этим расширением, откройте *LXF71* на 90 странице и прочитайте – вам откроется целый новый мир

4) Будьте объектно-ориентированными. Это было ужасно трудно в PHP 4, но начиная с PHP 5 web-разработчикам стало гораздо легче жить. Большая часть web-страниц прекрасно описывается в рамках объектно-ориентированной парадигмы – как минимум саму страницу можно рассматривать в виде объекта, с методами наподобие *writeHeader()*. Но если вы работаете с бизнес-логикой или создаете приложения, выполняющие конкретные задачи (например игры или графические интерфейсы

пользователя), то надо быть сумасшедшим, чтобы не использовать ООП.

5) Кэшируйте PHP-код. APC – это прекрасное (и бесплатное!) решение, позволяющее вашей программе выполняться гораздо быстрее. А если у вас есть свободные деньги, вы можете инвестировать их в покупку Zend *Platform*, благодаря которому вы получите самый быстрый PHP-код на свете.

6) Не стоит использовать PHP 4.x, *Apache 1.3.x* или *MySQL 4.0* или меньше, если, конечно, вашему серверу есть куда девать процессорное время. Все эти программы устарели уже на две версии, так что отсутствие обновлений уже непростительно. А если вы все еще не пользуетесь переменными **$_GET, $_POST** и так далее – прекратите прямо сейчас.

7) Не используйте MD5 в сценариях, которые должны быть криптографически-безопасными. Вместо него возьмите SHA1, благо повышение безопасности практически не вызывает потери производительности. Сейчас MD5 можно использовать исключительно для обратной совместимости.

8) Вместо функций *fopen()*, *fread()*, *fwrite()* и *fclose()* напишите просто *file_get_contents()* и *file_put_contents()*. Нет, производительность тут ни при чем, это просто вопрос удобочитаемости кода.

9) Не стоит слепо писать всюду *mysqli_connect()* и другие специфичные для базы данных функции. PDO – это исключительно быстрая, а в сочетании с Pear DB еще и очень гибкая система доступа к СУБД.

10) Пожалуйста, не используйте *eval()*. А если вы до сих пор продолжаете это делать – киньте мне email, что бы я мог вдоволь поиздеваться.

Что ждет нас впереди?

Одно из удовольствий в жизни программиста – это то, что его мир никогда не остается неизменным. Хотя нельзя сказать, что PHP движется со скоростью света, но каждый год вы вполне можете ожидать новый релиз, а так же множество обновлений библиотек и появление новых сценариев. Изучение этого всего не дает потерять интерес к работе, так что вам приятно будет узнать, что за горизонтом нас ждет еще больше неизведанного.

Для начала, в Perl 6 будет полностью переработан синтаксис регулярных выражений. Пока это изменение нас не касается, и все же нам не стоит о нем забывать, поскольку, во-первых, в PHP используется библиотека совместимых с Perl регулярных выражений, и мы не знаем, когда из совместимой с Perl 5 она превратится в совместимую с Perl 6, и во-вторых новый синтаксис выглядит гораздо лучше. Один из девизов Perl – «Простые вещи должны быть простыми, а сложные вещи должны быть возможными». Новый синтаксис разработан так, чтобы все основные конструкции было проще набрать, а это значит – быстрее обработать, проще прочитать и легче выучить.

Второе, о чем стоит думать – это увеличение роли PHP как языка для связывания объектов бизнес-логики (middleware). Да, PHP всегда неплохо подходил для такого связывания, но все же он имел репутацию детской игрушки, не пригодной к использованию в серьезной компании. Это мнение пришлось опровергать с двух сторон: придавая PHP более Java-подобный синтаксис и добавляя дополнительные зацепки в Java, чтобы вызывать бизнес-объекты было проще. Со старым добрым набором технологий LAMP PHP-программист всегда мог заработать себе на хлеб с маслом. Но не забывайте смотреть дальше – и там вас ждут настоящие деньги!



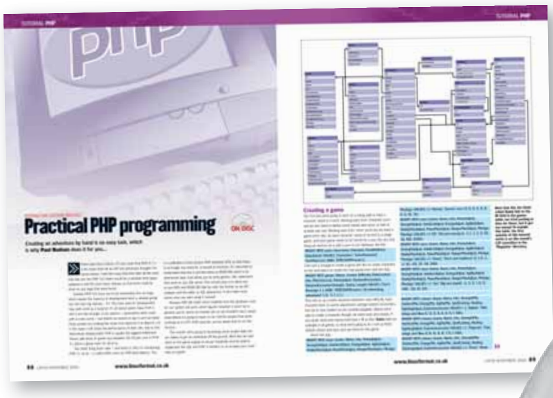
Что наша жизнь? Игра! И PHP в ней выигрывает.

Ну и наконец, сейчас вам действительно стоит потратить время на изучение стандартной библиотеки PHP. Это встроенный набор объектов для выполнения повседневных дел, а так же обработчики специфических задач, таких как проход по всем файлам каталога, и более общих алгоритмов наподобие перебора всех элементов обычного массива. Эта библиотека становится стандартом в PHP, особенно вместе с хитрыми расширениями вроде SimpleXML. По-моему, стандартная библиотека будет следующим большим событием в PHP, так что просто будьте на пике моды.

Долгое прощание

Ну, вот и все. Если вы прочитали и попробовали все эти статьи – спасибо вам, я надеюсь что вы много узнали и теперь полностью вооружены для более сложных проектов. Если вы пропустили что-то – вы можете вернуться и прочитать пропущенное с DVD. В любом случае, на этом моя серия заканчивается, но в вашей карьере PHP-программиста все только начинается (тут можно сказать много других стандартных напутствий).

Поздравляю, вы закончили школу программирования Пола Хадсона! Идите, и захватите мир, уничтожьте программистов JSP и сделайте сервера *Apache* счастливыми до скончания веков. Прощайте! **LXF**

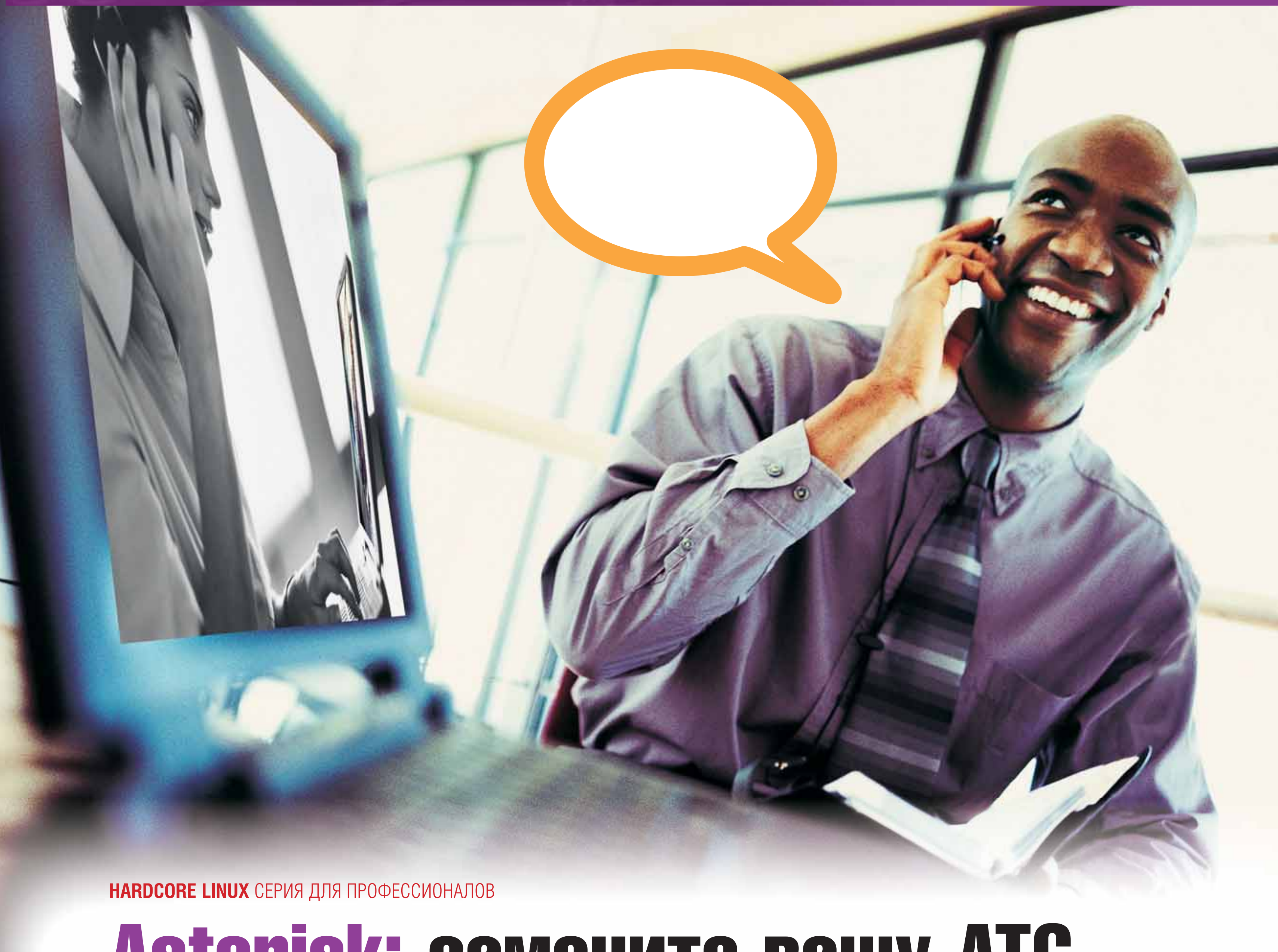


К *LXF59* я стал платить нашим художникам достаточно для того, чтобы они могли рисовать такие вот таблички.



ПОДСКАЗКИ

• Если вы действительно хотите еще, то посмотрите книгу «*Практическое PHP-программирование*» (*Practical PHP Programming*). Она бесплатно доступна в Интернете по адресу www.hudzilla.org/phpbook. А если вы нуждаетесь в совете или обсуждении – подумайте о вхождении в группу рассылки пользователей PHP.



HARDCORE LINUX СЕРИЯ ДЛЯ ПРОФЕССИОНАЛОВ

Asterisk: замените вашу АТС

Душа горит желанием внедрить в своей организации VoIP взамен недешёвой АТС, но не уверены, что справитесь? Попробуйте эту открытую систему и добавьте отличные функции – например, конференции с интернет-вызовами.

НА ДИСКЕ

МЕСЯЦ НАЗАД



В прошлом выпуске вы вонзили зубы во множество проектов, но в *LXF80* Крис Браун провёл мастер-класс по регулярным выражениям.

Выход технологии за пределы круга своих приверженцев в коммерческий мир всегда волнителен. Это произошло с Интернетом в середине девяностых; с Linux незадолго до смены тысячелетий; и теперь это происходит с IP-телефонией.

Поскольку «голос поверх IP» (Voice over IP, VoIP) – это сервис, работающий поверх общедоступного Интернета (или, возможно, сегмента частной LAN/WAN), большинство протоколов и стандартов, используемых данной технологией, открыты, и это способствует разработке широкой номенклатуры программ и оборудования. Есть несколько различных реализаций VoIP, каждая из которых имеет свои преимущества и недостатки.

Приложение, которое я здесь демонстрирую, *Asterisk* – это открытая УАТС (учрежденческая АТС), или телефонный узел частного пользования (private branch exchange, PBX). Хотя *Asterisk* был разработан для использования с VoIP – и поддерживает богатый набор протоколов VoIP, включая SIP и H.323 – им вполне можно бесплатно заменить УАТС даже и без функций VoIP. Мы особо приглашаем собрать её для себя любителей удушающе больших счетов от телекоммуникационных компаний и непристойных тарифов на «дальние» и международные звонки (см. врезку «*Подключение к телефонной сети общего пользования*»).

На этом уроке я предполагаю, что вы пошли путём свободного VoIP. Мы сосредоточимся на трёх специфических функциях *Asterisk*:
- Как уберечь передачу вызовов от нестабильности Интернета.
- Как поддерживать связь с друзьями и коллегами с помощью конференции.
- Как помещать входящие звонки в очередь

Но как это работает? Кто использует УАТС? И о чём я вообще тут толкую?

Если вы не привыкли к телефонии, вам придётся потерпеть, пока я коротко опишу её функционирование – но, с другой стороны, крутые хакеры вроде вас должны бы терпимо относиться к техническим разъяснениям. Приступим. Любая реализация VoIP в принципе состоит из двух компонентов: процесс, передающий звуковую информацию от пользователя в сеть VoIP (обычно, телефон), и процесс, позволяющий оборудованию VoIP взаимодействовать с телефонной сетью общего пользования (ТфОП). Хотя люди с техническим складом ума из многих

РАСШИФРОВКА ТЕРМИНОВ VOIP

- **E1** – Транспортная шина, используемая в Европе, состоит из 32 каналов по 64 кбит. Каналы могут использоваться для передачи данных, голоса или и того, и другого.
- **Foreign eXchange Office (FXO)** – Интерфейс, используемый для подключения к вашей местной телефонной компании по стандартной ТфОП-линии (PSTN). Он называется FXO, поскольку через него мы подключаемся к центральному офису нашего оператора.
- **Foreign eXchange Station (FXS)** – Интерфейс, используемый для подключения стандартных телефонных аппаратов. Телефонная розетка в вашей стене.
- **G.711** – Стандартный метод кодирования, используемый телефонными компаниями для передачи цифровых сигналов. В Европе применяют G.711 «alaw», в США – G.711 «ulaw».
- **G.729** – Узкополосный кодек, который запатентован, так что за его использование приходится платить. Но зато при использовании G.729 есть возможность передавать вызовы VoIP по модему 56 кбит.
- **H.323** – Сравнительно старый протокол, использовавшийся для голосовых коммуникаций. Большинство устройств, которые использовали H.323, теперь вместо него используют SIP.
- **Inter-Asterisk eXchange (IAX)** – Открытый протокол VoIP от Digium, основного разработчика *Asterisk*, спроектированный для надёжной передачи голосовых вызовов через IP. IAX хорошо подходит для магистральных вызовов между оператором VoIP и локальной системой или между двумя системами *Asterisk*.
- **Джиттер (Jitter)** – Общая проблема VoIP, когда голосовые пакеты задерживаются на различное время, и абонент слышит паузы или нарушение порядка звуковых фрагментов в последовательности.
- **Real-time Transmission Protocol (RTP)** – Протокол передачи, работающий поверх UDP и обеспечивающий взаимодействие в режиме реального времени по IP. RTP используется как в SIP, так и в H.323 для потоков голосовой информации.
- **Session Initiation Protocol (SIP)** – Популярный протокол VoIP, используемый как в телефонах, так и операторами; он действительно хорош, за исключением работы через некоторые реализации NAT.
- **T1** – транспортная шина, используемая в США для передачи голоса или данных. T1 состоит из 24 каналов по 64 кбит, используемых для входящих или исходящих вызовов. Также поверх T1 может передаваться ISDN PRI для 23 голосовых каналов.

стран терпеть не могут общую телефонную сеть, всё же это стандарт, и к ней подсоединён почти каждый. Как бы ни была хороша технология VoIP, пользуются ею не все, так что не обойтись без способности звонить на стандартные телефонные системы и принимать вызовы практически от любого абонента в мире.

Наша задача согласовать две отдельные телефонные сети не нова – любое предприятие с более чем пятью сотрудниками, вероятно, имеет внутреннюю телефонную сеть с расширениями и голосовыми услугами. Чтобы подключить эту сеть к общей коммутационной сети (public switched telephone network, PSTN), используется УАТС. УАТС можно рассматривать как телефонный маршрутизатор, со своим внутренним диапазоном адресов (известных также как добавочные номера или расширения, extensions) и доступом к общим номерам снаружи. УАТС хорошего качества стоят не дёшево (несколько тысяч долларов и более за абсолютный минимум функций), зато предоставляют ожидаемую пользователями надёжность почти со 100%-й готовностью на протяжении нескольких лет, если не десятилетий.

При формировании среды VoIP, для ИТ-персонала УАТС – первоочередная область внимания, поскольку это линия фронта между ТфОП и нашими пользователями. Коммерческие системы УАТС с поддержкой VoIP предлагаются почти всеми, кто использует стандартные устройства УАТС – посмотрите на решения Nortel (**www.nortel.com**), Lucent (**www.lucent.com**) и Cisco (**http://cisco.com**) – но имеется замечательная альтернатива в настоящем стиле open source, работающая на стандартном оборудовании Intel. И здесь мы возвращаемся к *Asterisk*.

Будем знакомы

Эта программа была разработана Марком Спенсером [Mark Spencer], чья телефонная компания Digiun оказала проекту разработке *Asterisk* первичную поддержку, а вообще-то поставляет телекоммуникационное оборудование. *Asterisk* бесплатен для загрузки, что делает использование УАТС возможным для очень маленьких фирм и даже отдельных пользователей.

Во многих дистрибутивах, включая Fedora Core и Debian, доступны двоичные пакеты *Asterisk*, а также модули ядра и требуемые библиотеки поддержки. Пока оборудование для соединения с ТфОП (PSTN) не используется, поддержка на уровне ядра не нужна. Пользователи Debian могут просто выполнить **apt-get install Asterisk**, чтобы загрузить и установить систему *Asterisk*, или скачать исходные коды с **www.Asterisk.org**.

Сборка *Asterisk* проста, хотя и требует компиляции *libpri* и *zaptel* для C-заголовков. Установить его можно стандартно: **make && make install** с правами суперпользователя – для большинства пользователей сборка пройдёт успешно. *Asterisk* потребует немного времени на компиляцию, но, будучи установленным, он будет использовать некоторые файлы конфигурации по умолчанию, поставляемые с пакетом.

Мы можем протестировать установленный нами *Asterisk*, выполнив от имени суперпользователя:

```
# Asterisk -cvvvvv
```

На экране замелькает всевозможная информация, по мере загрузки каждого модуля, но в конечном счёте мы увидим простенькое приглашение, с помощью которого сможем управлять нашей системой *Asterisk*:

```
Asterisk Ready.
CLI>
```

Командная строка *Asterisk* весьма напоминает *Cisco IOS*, так что если вы знакомы с этим популярным ПО для маршрутизаторов, то моментально в ней разберётесь. Команда *show version* выведет текущую

> «ASTERISK ПРЕДОСТАВЛЯЕТ НАМ ГИБКОСТЬ ДЛЯ РАБОТЫ С БОЛЕЕ ЧЕМ ОДНИМ ОПЕРАТОРОМ VOIP.»

версию *Asterisk*, а все команды можно просмотреть, введя **?** в командной строке. Поначалу наиболее полезной командой будет *show modules*, которая отображает все модули, входящие в комплекс *Asterisk*. Каждая из возможностей *Asterisk*, например, обеспечение протоколов телефонии, кодеки для сжатия звука и различные устанавливаемые по умолчанию приложения, представлена в *Asterisk* в виде модуля.

Устойчивая маршрутизация вызовов

Теперь, после установки, самое время воспользоваться *Asterisk* для решения нашей первой общей проблемы телефонии: непрофессиональная маршрутизация вызовов.

Понятно, что полным-полно провайдеров VoIP с бизнес-моделями отнюдь не звёздного уровня (см. врезку «Выбор вашего оператора VoIP», там перечислены операторы, имеющие репутацию солидных). Пусть они и предлагают звонки в Тибет за копейки, но человек разумный вряд ли будет ожидать высокой стабильности их сети. *Asterisk* пре- >>

ВЫБОР ВАШЕГО ОПЕРАТОРА VOIP

Если вы планируете использовать *Asterisk* с сервисом VoIP, чтобы звонить через интернет, вам потребуется подключиться к оператору, который сможет перенаправлять ваши звонки. Проверьте:

- **Gossiptel** – www.gossiptel.co.uk
- **Gradwell** – www.gradwell.net
- **NuFone** – www.nufone.net
- **Simple Telecom** – www.simpletelecom.co.uk
- **VoIP User** – www.voipuser.org

Есть также несколько проектов сообщества, которые не умеют перенаправлять звонки в сеть ТфОП, но удобны для соединения в Интернете с другими владельцами систем VoIP:

- **Free World Dialup** – www.freeworlddialup.com
- **IAXTel** – www.iaxtelcom

Множество подробной информации о провайдерах VoIP можно найти на www.voip-info.org/wiki-VOIP+Service+Providers.

<< доставляет достаточную гибкость в подключении к более чем одному оператору VoIP для обработки вызовов, и мы можем построить логику нашей УАТС так, что если один оператор вдруг «умрёт», для звонков во внешний мир станет использоваться другой метод.

Жизненно важная составляющая в этой смеси – приведённый ниже макрос «дозвона», он важнее всех тех, что вам доведётся увидеть (по крайней мере, сегодня на уроке):

```
[macro-dial]
exten=>s,1,Dial(${ARG1},120)
exten=>s,2,Goto(s-${DIALSTATUS},1)
exten=>s-NOANSWER,1,Handup
exten=>s-BUSY,1,Busy(45)
exten=>s-CONGESTION,1,NoOp
exten=>s-CHANUNAVAIL,1,NoOp
exten=>s-.,1,Goto(s-NOANSWER,1)
```

Как видите, макрос позволяет нам создать список методов для передачи клиентских звонков и не волноваться об «уборке» после себя.

Мы можем реализовать этот макрос в соответствии с нашим номерным планом и логикой, используемой для перенаправления вызовов, в **extensions.conf**:

```
exten=>_0.,1,Macro(dial,IAX2/carrier1/${EXTEN})
exten=>_0.,2,ResetCDR
exten => _0.,3,Macro(dial,IAX2/carrier2/${EXTEN})
```

«ИСПОЛЬЗОВАНИЕ АГЕНТОВ ПОЗВОЛЯЕТ ВСТАВАТЬ В ОЧЕРЕДЬ, КОГДА НУЖНО ПРИНИМАТЬ ЗВОНКИ»

```
exten=>_0.,4,Hangup()
```

Эту конфигурацию можно расширять как угодно, добавляя в список других операторов VoIP. Её недостаток – когда звонок уже принят, нет гарантии, что он пройдёт гладко. Немногие конечные пользователи хотели бы иметь дело с недостатками вроде плавающего или одностороннего звука и прерванных соединений, но как только провайдер VoIP берёт вызов в свои руки, мы, естественно, предполагаем, что он всё сделает правильно. Если у вас достаточно времени, вы, думаю, сможете набросать что-то в пять строк на Perl для анализа вызовов, сделанных из *Asterisk*, и определить, какие маршруты более надёжны.

В больших системах основное преимущество даёт маршрутизация по критерию наименьшей стоимости (Least Cost Routing, LCR), разновидность услуги сравнения, особенно когда у провайдеров различаются тарифы для звонков за границу. Тарифные планы пары провайдеров можно поместить в базу данных и использовать для определённых звонков провайдера, предложившего более низкую цену. Подробную информацию можно получить на http://cpan.uwinnipeg.ca/htdocs/Asterisk-LCR.

Строим конференц-связь

Теперь – наша вторая задача. Все телефоны, кроме самых простых, способны принимать трёхсторонние звонки, так что можно организовать небольшую конференцию с двумя другими абонентами. Если участников больше трёх, это усложняется и требует усовершенствованного решения, ориентированного на группы абонентов. Даже если вы проводите основную часть своих групповых дискуссий в сети, не мешает знать, как создать мост конференций и запустить его, на случай, если сеть IRC будет неработоспособна.

Для поддержки телефонных конференций *Asterisk* предоставляет приложение, известное как *MeetMe*; примеры конфигураций включены в поставку *Asterisk*. Основная зависимость *MeetMe* – подсистема времени *Asterisk*. Обычно, чтобы предоставить точные часы, *Asterisk* использует драйвер Zaptel и физические устройства; однако это доступно не всегда, особенно в случае серверов-стоек, где разъёмы PCI в большом дефиците. Вместо этого, в ядре 2.6 и текущем релизе Zaptel вы можете использовать модуль ядра ztdummy, и всё замечательно заработает. Многие функции *Asterisk* связаны с синхронизацией, так что при подгруженном *ztdummy* всё работает гораздо более гладко.

Построить мост конференций в *MeetMe*.conf совсем не сложно, требуется только номер ID конференции и необязательный PIN-код для доступа:

```
conf => 2345,9938
```

Мы можем затем переключиться на этот мост из нашего **extensions.conf**:

```
exten => 2000,1,MeetMe(2345)
```

Первому позвонившему сообщат, что он пока только один, и будет проигрываться мелодия ожидания. Как только присоединится второй абонент, мелодия ожидания завершится, и они смогут побеседовать. При подключении или отсоединении очередного абонента все будут оповещаться об этом звуковым сигналом. *MeetMe* имеет огромный набор опций, включая расширения для подключения к конференции с различными предпочтениями. Это идеально, когда мы хотим, чтобы определённые абоненты подсоединились и послушали, но не смогли ввязаться в разговор; или когда к мосту уже подключено несколько человек, но мелодия ожидания звучит до тех пор, пока не объявится некая конкретная личность.

Есть ряд версий мостов конференций сторонних разработчиков, работающих с *Asterisk*, но *MeetMe* имеет так много опций, что его замену обосновать очень трудно.

Управление очередью

Теперь мы можем заняться третьей задачей *Asterisk*, которая ждёт своего часа, слушая мелодию Greensleeves буквально с самого начала урока. Любая организация, клиентов у которой намного больше, чем сотрудников на телефоне, должна рационально расставить приоритеты звонков и обрабатывать их так, чтобы люди не спятили окончательно. Очереди звонков хороши для решения первой проблемы, хотя некоторые считают, что они могут серьёзно подорвать психическое здоровье клиентов. Типичная очередь – это система, когда дозвонившийся абонент слушает



некоторое время музыку, затем один из агентов может поднять телефон и ответить на звонок. *Asterisk* легко с этим справляется при помощи приложения Queue.

Очереди определены в файле **queues.conf**, который имеет ту же структуру, что и другие файлы конфигурации *Asterisk*. Каждая очередь строится с собственной конфигурацией и списком агентов. Простейшая очередь может выглядеть примерно так:

```
[support]
musiconhold=default
strategy=roundrobin
timeout=15
retry=5
wrapuptime=30
maxlen=0
announce-frequency=90
announce-holdtime=yes
announce-round-seconds=60
context=operator
reportholdtime=yes
member => SIP/200
member => SIP/208
member => SIP/212
```

Большая часть этой конфигурации довольно прямолинейна, хотя есть ряд опций, подстраиваемых под требования конкретной очереди. Строки **announce** позволяют приложению периодически сообщать звонящему, сколько он просидел в очереди и какова его позиция, и радовать его позорными сообщениями типа: «Вы – следующий на очереди, среднее время разговора – пять часов». Опция **reportholdtime** включает уведомления агенту о том, как долго звонящий ждал соединения. Отделам технической поддержки такая информация полезна для определения тенденций по времени вызова и может быть включена в документацию по запросу.

В нашей базовой конфигурации очереди есть только статические агенты, которые не могут покинуть очередь, пока не переведут свои телефоны в режим DnD (Do not Disturb, «не беспокоить») или не отключатся. Для людей, которые приходят и уходят, или для удалённых пользователей использование агентов позволяет подключаться к очереди, когда они хотят ответить на звонок. Система отсоединит их, если они не смогут ответить в течение определённого времени. Создадим **agents.conf**, похожий на этот:

```
[agents]
ackcall=no ; Агент не нажал #, чтобы ответить на звонок
musiconhold => default
agent => 1234,0000,Agent1_Name
agent => 1235,0000,Agent2_Name
```

Для подключения и отсоединения используется команда *AgentCallBackLogin* в файле **extensions.conf**, чтобы предоставить расширение для аутентификации пользователей по их ID и PIN-коду, прежде чем они присоединятся к очереди в качестве агента:

```
exten => 700,1,AgentCallBackLogin(|${CALLERIDNUM}@local)
```

В этом примере вызовы для очереди будут выполняться на идентификатор подключённого лица, исключая необходимость определять его местоположение. Однако имейте в виду: если ваш телефонный номер снабжен добавочным, идентификатор может оказаться неправильным местом для перенаправления вызова.

Узнайте больше

Нарождающаяся технология развивается с необычайной быстротой – за ней трудно уследить. Сайт www.voip-info.org – чудесный ресурс для каждого, кто работает с VoIP, даже если *Asterisk* не является «частью уравнения». Даже не пытайтесь настроить VoIP-телефон от Cisco или Polycom, не посетив этот сайт: без толкового руководства вы можете зря потратить время, пытаясь во всём разобраться. LXF

СОВЕТЫ

- На загруженной системе Asterisk не пяльтесь зря на командную строку управления (CLI). Каждое сообщение, отправленное на CLI, приводит к тому, что Asterisk блокирует свою систему обработки вызовов, задерживая новые звонки.
- Команда show может предоставить много полезной статистики: show queues для очередей, show channels для обрабатываемых звонков, и т.д.
- После изменения опций команда reload перезапустит Asterisk без сброса установленных соединений. Это очень полезно при редактировании добавочных телефонных номеров или модификации очередей.
- Запуск iax2 show channels предоставит информацию по задержкам и помехам для каждого активного вызова IAX2, бесценную для поиска неисправностей сети.

ПОДКЛЮЧЕНИЕ К ТЕЛЕФОННОЙ СЕТИ ОБЩЕГО ПОЛЬЗОВАНИЯ

Интернет ненадёжен? Звоните через старую добрую сеть общего пользования (PSTN) и вашего телефонного провайдера.

Digium предлагает порт FXO для подключения *Asterisk* к обычной линии **PSTN**, используя карту **X100P** или плату **TDM400P** и FXP-модуль. Карта **X100P** предоставляет одни порт FXO, в то время как **TDM400P** позволяет подключать 4 порта, настраиваемые как FXO, либо FXS (для телефонных аппаратов), используя маленькие подсоединяемые модули.

Первая карта довольно дорога – примерно 180 долларов, и поставляется с одним портом FXS; однако поскольку она использует лишь один разъём PCI и отличается столь значительной гибкостью, **X100P** при подключении к **PSTN** – это путь наименьшего сопротивления. Оба устройства поставляются с техподдержкой в установке оборудования и настройке *Asterisk*, так что вы сможете получить профессиональный совет, если что-то пойдет не по плану.

Установив оборудование, нужно будет загрузить соответствующие модули ядра и настроить интерфейсы. *Asterisk* для этих устройств использует систему под названием Zaptel, и ссылается на них как на Zap-каналы. Конкретные типы каналов настраиваются в **/etc/zaptel.conf**, где устанавливается тип порта FXO и FXS и указывается, в какой стране мы находимся.

Чтоб жизнь малиной не казалась, порт FXO использует протокол FXS, а порт FXS – протокол FXO. Так, даже если у нас установлен модуль FXO в канале 1 (порт 0), мы должны сообщить Zaptel, что на этом интерфейсе хотим действовать по FXS. Всё, что касается Zaptel, настраивается в **/etc/zaptel.conf**, который выглядит примерно так:

```
fxsks=1
fxoks=2
loadzone=us
defaultzone=us
```

Строки **fxsks** и **fxoks** сообщат оборудованию, что следует использовать метод сигнализации KewlStart, это стандартный протокол аналоговых телефонных линий (**POTS**) в западных странах для взаимодействия с коммутатором телефонной компании. Если порт FXO не исходит от банка каналов на магистральную линию, где сигнализация может придерживаться метода, зависящего от выбора, сделанного при подготовке линии, KewlStart – та опция, которая нам нужна.

Как только **/etc/zaptel.conf** будет создан, мы можем загрузить модули, настроить Zaptel и проверить, что всё работает как надо:

```
# modprobe zaptel wcfxs
# ztcfg
# dmesg
Zapata Telephony Interface Registered on major 196
Freshmaker version: 63
Freshmaker passed register test
Module 0: Installed -- AUTO FXO (FCC mode)
Module 1: Installed -- AUTO FXS/DPO
Module 2: Not installed
Module 3: Not installed
Found a Wildcard TDM: Wildcard TDM400P REV E/F (4 modules)
Registered tone zone 0 (United States / North America)
```

Чтобы обрабатывать исходящие и входящие вызовы, нужно настроить **/etc/asterisk/zapata.conf** и **/etc/asterisk/extensions.conf**.

```
signalling=fxs_ks
context=from-pstn
channel => 1
signalling=fxo_ks
context=internal
channel => 2
```

Любой входящий вызов будет подпадать под контекст from-pstn в **extensions.conf**, а исходящие звонки мы можем сбрасывать на канал Zap/1. Также доступен канал Zap/2, к которому можно присоединить стандартный телефон, как к стандартной телефонной линии. Получая входящий звонок на наш канал Zap/1, мы можем использовать стандартные телефоны, а также затейливую трубку Cisco SIP.



ЧЕРЕЗ МЕСЯЦ

Мы ещё глубже погрузимся в тайны Linux, рассмотрев подключаемые модули аутентификации, известные линуксоидам как PAM.



PYTHON ДЛЯ ПРОФЕССИОНАЛОВ

РАЗРАБОТКА клиент-серверных приложений

ЧАСТЬ 2 Вознамерились написать открытую альтернативу Skype или собственный клиент BitTorrent? **Сергей Супрунов** научит всему необходимому – от основ архитектуры «клиент-сервер» до готовых библиотек для работы с существующими интернет-протоколами.

МЕСЯЦ НАЗАД

Мы познакомились с ветвлением процессов, сокетами и модулем **select**.

НА ДИСКЕ

- Код примеров статьи

В прошлый раз мы начали разговор о многозадачности. А ведь возможности одновременно выполнять несколько задач наиболее востребованы при построении сетевых приложений, работающих по схеме «клиент-сервер».

Клиент всегда прав

Наиболее распространенным способом взаимодействия двух приложений через сеть являются уже знакомые нам сокеты. Модуль *socket*, входящий в стандартную поставку *Python*, помимо рассмотренных в прошлый раз Unix-сокетов поддерживает также сокеты домена Internet. Методология использования мало чем отличается от Unix-сокетов, за исключением того, что вместо параметра **AF_UNIX** используется **AF_INET**, а вместо имени файла указываются имя хоста и номер порта, которые будут обслуживаться создаваемым сокетом.

В качестве второго параметра в конструкторе сокета можно указать его тип: с установлением соединения, соответствующий протоколу TCP, или без соединения – протокол UDP. Допустимые значения – **SOCK_STREAM** и **SOCK_DGRAM** соответственно. По умолчанию подразумевается **SOCK_STREAM**.

Как пример, рассмотрим работу приложения, выполняющего роль примитивного (и весьма ограниченного функционально) клиента DNS, работающего по протоколу UDP (*см. листинг **udp-client.py***).

Некоторую сложность здесь представляет то, что DNS относится к так называемым «двоичным» протоколам, в отличие от «текстовых», таких как HTTP или SMTP, где обмен идет обычными текстовыми строками. В случае с DNS оперировать приходится «сырыми» байтами. Сведения по формату сообщений можно почерпнуть из RFC 1035, раздел «4. MESSAGES».

В 6-й и 7-й строках рассматриваемого кода формируется заголовок (12 байт). Пренебрегая всем богатством возможностей протокола DNS,

UDP-CLIENT.PY

```
1  #!/usr/bin/python
2  # -*- coding: utf-8 -*-

3  import sys
4  from socket import *

5  hostname = sys.argv[1]

6  HEADER = '\x00\x01\x00\x00\x00\x01'
7  HEADER += '\x00\x00\x00\x00\x00\x00'

8  QUESTION = ''
9  parts = hostname.split('.')
10 for p in parts:
11     QUESTION += '%c%s' % (chr(len(p)), p)
12 QUESTION += '\x00\x00\x01\x00\x01'

13 QUERY = HEADER + QUESTION

14 cs = socket(AF_INET, SOCK_DGRAM)
15 cs.sendto(QUERY, ('127.0.0.1', 53))

16 rsp = cs.recv(1024)

17 start = len(QUERY) + 12
18 print '%s.%s.%s.%s' % (ord(rsp[start]),
19                    ord(rsp[start+1]),
20                    ord(rsp[start+2]),
21                    ord(rsp[start+3]))
```

мы ограничиваемся простым запросом (**OPCODE=0**) одного доменного имени (**QDCOUNT=1**). Конструкции вида **«\x00»** позволяют задать в строке произвольный шестнадцатиричный код.

В строках 8–12 формируется поле запроса. Оно состоит из частей доменного имени (в оригинале разделенных точками), перед которыми указывается число символов в этой части. Например, имя mail.ru состоит из двух частей (mail – 4 символа, ru – 2 символа) и в запросе должно выглядеть так: **\x04mail\x02ru\x00**. Завершающий ноль, а также два двухбайтовых поля (QTYPE и QCLASS) добавляются к переменной **QUESTION** в строке 12.

Наконец, строки 14 и 15 – создание сокета (обратите внимание на второй параметр, **SOCK_DGRAM**, указывающий тип транспортного протокола UDP) и отправка запроса. Поскольку UDP работает без установки соединения, то вместо знакомой нам пары методов «connect – send» мы используем один «sendto», в котором указывается сразу и отправляемая информация, и адрес получателя. В данном примере предполагается, что DNS-сервер работает на локальной машине. Вы можете указать здесь свой DNS-сервер или передавать его имя в качестве параметра.

И в строках 17–21 из всей полезной информации, возвращаемой сервером, мы, игнорируя любые возможные ошибки, выбираем только IP-адрес, размещаемый по смещению, которое формируется в переменной **start**. Результат работы:

```
serg$ ./udp-client.py donpac.ru
80.254.111.2
```

Всегда к вашим услугам

В качестве примера сервера, на этот раз работающего по протоколу TCP, рассмотрим такой код (*см Листинг iamok.py*).

IAMOK.PY

```
#!/usr/bin/python
# -*- coding: utf-8 -*-

import os, re
from socket import *

class IamOK:
  def __init__(self, host='localhost', port=12345):
    self.socket = socket(AF_INET, SOCK_STREAM)
    self.socket.bind((host, port))
    self.socket.listen(5)

  def process(self):
    while 1:
      csocket, caddress = self.socket.accept()
      csocket.send('IamOK server v.0.0. Ready to serve.\n')
      csocket.send('You are from %s, port %s...\n' % caddress)
      while 1:
        request = csocket.recv(64)
        if re.match('get\s+uptime', request, re.IGNORECASE):
          csocket.send(os.popen('/usr/bin/uptime').read())
        elif re.match('quit|bye|exit', request, re.IGNORECASE):
          break
        else:
          csocket.send('Unknown command.\n')
      csocket.close()


if __name__ == '__main__':
  serv = IamOK()
  serv.process()
```

Думаю, вы уже поняли, что он прослушивает указанный порт (12345), и при поступлении на него запроса возвращает клиенту выводэ утилиты uptime, из которого можно почерпнуть время непрерывной работы сервера, число подключенных в данный момент пользователей и среднюю загрузку системы.

Здесь все должно быть понятно по прошлому уроку. Два отличия – в конструкторе **socket.socket()** указывается второй параметр – **SOCK_STREAM** (в данном случае его можно было бы и опустить, т.к. для Internet-домена и так по умолчанию используется протокол TCP). И метод **bind()** осуществляет привязку сокета не к файлу, а к имени хоста и номеру порта, на котором будут ожидаться входящие соединения. Кстати, выбирая номер порта, не забывайте, что порты до 1024-го относятся к привилегированным и могут быть задействованы только пользователем root.

Наш сервер понимает две команды: **«get uptime»**, по которой возвращается информация о времени работы сервера, и **«quit»** (с двумя синонимами – **«bye»** и **«exit»**), по которой сеанс завершается.

Проверить работу сервера можно с помощью обычной telnet-сессии:

```
admin@dom:~/lxf/propy/l2$ telnet localhost 12345
Trying 127.0.0.1...
Connected to localhost.localdomain.
Escape character is '^]'.
IamOK server v.0.0. Ready to serve.
You are from 127.0.0.1, port 2650...
helo
Unknown command.
get uptime
 23:09:26 up 58 min, 3 users, load average: 0.04, 0.15, 0.63
quit
Connection closed by foreign host.
```

Какая от этого может быть польза – решайте сами.

«Гуртом и батьку бить веселей»

Потоки, наряду с рассмотренными ранее процессами, являются эффективным способом параллельного выполнения различных задач. В ряде случаев этим можно воспользоваться не только для одновременного обслуживания нескольких подключений в клиент-серверных приложениях, но и для повышения производительности «автономных» программ.

Например, пусть у нас есть каталог с десятком достаточно больших журнальных файлов. Их требуется обработать, подсчитав число строк, в которых встречаются сообщения об ошибках. Данная задача создает нагрузку как на дисковую подсистему, так и на процессор. Но при последовательной обработке ресурсы будут расходоваться неэффективно – пока выполняется чтение очередного файла, процессор простаивает. Напротив, во время обработки считанных данных бездействует диск. Таким образом, здесь есть потенциал для оптимизации за счет обработки каждого файла в отдельном потоке.

В стандартной поставке *Python* для работы с потоками есть два модуля – *thread* и *threading*. Первый позволяет управлять потоками на достаточно низком уровне, второй – использует средства первого для предоставления более удобного объектно-ориентированного интерфейса. Класс *threading.Thread* предоставляет «шаблон» потока. Для его использования в своей программе вам нужно переопределить метод **run()**, описав в нем действия, которые должны выполняться этим потоком. Ниже приведен пример, из которого все должно стать понятно.

Нужно заметить, что при распараллеливании «в лоб», когда просто запускается сразу несколько потоков, по одному на каждый обрабатываемый файл, нас поджидает один неприятный сюрприз (можете проверить это на досуге) – при старте сценария все созданные потоки одновременно попытаются выполнить чтение нужных им для дальнейшей работы данных. Это приведет к жесткой конкуренции за право переместить головки винчестера в нужное место и вызовет, вопреки ожидаемому, катастрофическое падение производительности. Если размеры файлов сопоставимы с объемом оперативной памяти, система, ко всему прочему, может «впасть в своппинг», когда считанные одним потоком данные будут тут же записываться на диск, чтобы освободить место для данных другого потока.

Вот если бы нам удалось организовать работу потоков таким образом, чтобы, пока один из них выполняет чтение файла, другие не обра- >>

щались к диску... Для решения этой задачи в языке *Python* доступно несколько средств синхронизации работы потоков.

Простейшее из них – использование обычных переменных-флагов. Как вы помните, потоки разделяют оперативную память, принадлежащую процессу, в рамках которого они исполняются, так что изменения переменных будут «видны» всем процессам. Идея здесь проста:

```
if DISKBUSY:
  # ожидание
else:
  DISKBUSY = 1
  # чтение файла
  DISKBUSY = 0
```

То есть первый поток выставит истинное значение глобальной переменной **DISKBUSY** и приступит к чтению файла. «Опоздавшие» потоки будут ждать, пока переменная вновь не примет значение «ложь».

В такой реализации программисту предстоит решить не такую уж простую, как может показаться на первый взгляд, задачу – грамотно обеспечить ожидание. Бесконечный цикл проверки значения переменной слишком сильно нагружает процессор (не даром такие циклы называют напряженными). Напрашивающееся **time.sleep(1)** очень не эффективно – если ресурс освободится до того, как истечет время «спячки», то он будут простаивать. [Кроме того, подобный метод синхронизации сам по себе не атомарен – подробности ищите в статье «Очереди сообщений и семафоры» на стр. 102]

Однако в модуле *threading* есть готовая реализация описанной выше идеи – класс *Lock*, который предоставляет программисту так называемые блокировки, иногда именуемые «мьютексами» (mutex). Идя проста – создается объект данного класса (**threading.Lock()**), который имеет два метода: **acquire()** позволяет захватить объект, **release()** – освободить его. Метод **aquire()** является блокирующим: очередной поток, вызвавший его, будет ждать до тех пор, пока мьютекс не освободится.

Дальнейшим развитием идеи блокировок являются семафоры. Фактически, семафор – это тот же мьютекс, но позволяющий захватить себя несколько раз. Если вы создадите семафор командой **semaphore = threading.Semaphore(3)**, то его смогут захватить (тем же методом **semaphore.acquire()**) одновременно три потока (каждый раз отнимая по единице из указанного при инициализации числа). Четвертый поток сможет захватить семафор только после того, как он будет высвобожден (**semaphore.release()**) одним из тех, которые удерживают его в настоящее время.

Впрочем, для нашей задачи лучше всего подходят мьютексы – диск объявим неразделяемым ресурсом, и посмотрим, какой выигрыш по времени это нам даст (*см. листинг* ***thread-test.py***).

Здесь мы проводим два теста – последовательная обработка (1) и использование потоков с эксклюзивным доступом к диску (2). В строках 16–22 мы создаем подкласс класса *Thread*, в котором переопределяем метод **run()**. Запуск потока выполняется в строке 37. Обратите внимание на список **running** (строки 34, 38, 39–40). С его помощью мы отслеживаем активность потоков – метод **join()** заставляет ждать, пока поток не завершит свою работу. Дальнейшая работа основного сценария продолжится только после того, как отработают все порожденные потоки.

В строке 5 мы создаем мьютекс, с помощью которого в строках 8 и 11 будет регулироваться доступ потоков к диску. В глобальной переменной errors ведется подсчет числа строк, в которых есть подстрока «failed» или «error», в total – общее число обработанных строк. Результат работы:

```
21931 2302755 Послед.: 33.271387
21931 2302755 Потоки:  22.867245
```

Как видите, мы получили выигрыш по времени более чем на 30%. Но нужно заметить, что распараллеливание подобных скриптов дает заметный эффект только в том случае, если нагрузка на дисковую систему сопоставима с нагрузкой на процессор. Если какой-то из ресурсов будет востребован намного больше второго, то потокам все равно придется ждать его высвобождения, а с учетом дополнительных затрат на обслуживание самих потоков, суммарный результат может оказаться даже хуже, чем при последовательной обработке.

THREAD-TEST.PY

```
1  #!/usr/bin/python
2  # -*- coding: utf8 -*-

3  import threading as t
4  import time, re

5  diskbusy = t.Lock()

6  def parseit(lognum):
7      global errors, total
8      diskbusy.acquire()
9      log = open('logs/syslog.%d' % lognum)
10     lines = log.readlines()
11     diskbusy.release()
12     for line in lines:
13       if re.search('failed|error', line):
14         errors += 1
15       total += 1

16 class ParseLog(t.Thread):
17   def __init__(self, num):
18     self.lognum = num
19     t.Thread.__init__(self)
20
21   def run(self):
22     parseit(self.lognum)


23 #------------------ 1

24 def test1():
25   global errors, total
26   errors = total = 0
27   for i in range(10):
28     parseit(i)
29   print errors, total,

30 #------------------ 2

31 def test2():
32   global errors, total
33   errors = total = 0
34   running = []
35   for i in range(10):
36     tr = ParseLog(i)
37     tr.start()
38     running.append(tr)

39   for tr in running:
40     tr.join()
41
42   print errors, total,

43 start = time.time()
44 test1()
45 print 'Послед.: %f' % (time.time() - start)

46 start = time.time()
47 test2()
48 print 'Потоки:  %f' % (time.time() - start)
```

Все включено

В поставку *Python* входит несколько готовых модулей, позволяющих легко и быстро разработать сетевую программу, например, HTTP-сервер или FTP-клиент. Более детально вы сможете познакомиться с ними в документации или в хорошо прокомментированных исходных кодах самих модулей. Здесь же рассмотрим их возможности обзорно.

HTTP-SERVER.PY

```
1  #!/usr/bin/python
2  # -*- coding: utf8 -*-

3  import SimpleHTTPServer as http

4  handler = http.SimpleHTTPRequestHandler
5  server = http.BaseHTTPServer.HTTPServer(('localhost', 8080), handler)
6  server.serve_forever()
```

HTTP

Для работы с HTTP *Python* предоставляет четыре основных модуля: **BaseHTTPServer**, **SimpleHTTPServer**, **CGIHTTPServer** и **httplib**. Первые три реализуют простейшие серверы, причем второй и третий модули используют возможности первого, предоставляя программисту более высокоуровневый интерфейс к его методам. Модуль **httplib** служит для разработки HTTP-клиентов.

Например, простейший HTTP-сервер может выглядеть таким образом (*см. листинг* ***http-server.py***).

Как видите – всего четыре «рабочих» строчки, и то строка под номером 4 служит лишь для присвоения столь длинного имени метода-обработчика более короткой и удобной переменной. Вести себя этот сервер будет как «самый настоящий»: он будет возвращаться запрошенные html-страницы или файлы из текущего и вложенных в него каталогов, при наличии файлов **index.html** или **index.htm** в каталоге, из которого сервер запущен, клиенту по умолчанию (когда указано только имя каталога) будут отдаваться они. Если индексные файлы отсутствуют, автоматически будет строиться страница-содержание каталога (аналогично работает Apache с включенным модулем **mod_autoindex**). В ответ на запрос несуществующего ресурса будут возвращаться сообщения об ошибке, и т.д.

Клиентское приложение будет не намного сложнее (*см Листинг http-client.py*).

В строках 4–7 формируется нужный HTTP-заголовок, затем получаем и распечатываем ответ сервера:

```
admin@dom:~/lxf/propy/l2$ ./http-client.py
200 OK
<HTML><HEAD>
    <TITLE>Test page</TITLE>
</HEAD><BASE>
    <H2>It is a test page</H2>
</BASE></HTML>
```

Конечно, чтобы представить эту страницу в графическом отформатированном виде, придется приложить еще немало усилий. Но это, как говорится, уже дело техники.

HTTP-CLIENT.PY

```
1  #!/usr/bin/python
2  # -*- coding: utf8 -*-

3  import httplib

4  host = httplib.HTTP('localhost:8080')
5  host.putrequest('GET', '/testpage.html')
6  host.putheader('accept', 'text/html')
7  host.endheaders()

8  code, msg, headers = host.getreply()

9  print code, msg
10 if code == 200:
11   print host.getfile().read()
```

FTP-CLIENT.PY

```
#!/usr/bin/python
# -*- coding: uft-8 -*-

import ftplib

ftp = ftplib.FTP('ftp.freebsd.org')
ftp.login('ftp', 'my@mail.ru')
ftp.cwd('pub/FreeBSD')
retfile = 'README.TXT'
ftp.retrbinary('RETR %s' % retfile,
      open(retfile, 'w+').write, 1024)
ftp.quit()
```

Электронная почта

Модули **smtplib**, **poplib**, **imaplib** предоставляют клиентские интерфейсы к соответствующим протоколам. Их использование не намного сложнее рассмотренного выше **httplib**, и, думаю, вы без труда в них разберетесь. Для более тонкой обработки содержимого почтовых сообщений (выделения заголовков, вложений и т.д.) вам помогут модули **rfc822**, **mimetools**, **multifile**, **base64**, **mailbox** и другие. Все они очень хорошо прокомментированы и снабжены достаточно подробной документацией. Простейший способ получить к ней доступ – функция **help()**. Например:

```
>>> import mailbox
>>> help(mailbox)
```

Этот код выведет встроенную справку по работе с модулем **mailbox** прямо в окне интерактивного терминала.

FTP

Для работы с протоколом FTP к вашим услугам модуль **ftplib**. Работа с ним ведется на достаточно низком уровне, и порой напоминает обычный сеанс FTP, выполняемый вручную (*см. листинг* ***ftp-client.py***).

В итоге выполнения этого скрипта в текущем каталоге должен появиться файл **README.TXT**, скачанный с ftp-сервера ftp.freebsd.org.

Заключение

Итак, на этом мы завершим знакомство с основными сетевыми возможностями языка *Python*. Хочу заметить, что они выходят далеко за рамки простейших сценариев, пригодных для тестирования «больших» серверов или встраивания некоторых сетевых возможностей в ваши приложения. Приведу лишь несколько примеров. Так, в 90-х годах большой популярностью пользовался web-браузер *Grail*, разработанный на *Python* и предоставляющий весьма широкие для того времени возможности по обработке интернет-страниц – полная поддержка стандарта HTML 2.0 и, в значительной мере, HTML 3.2, поддержка различных форматов изображений и звука, способность работать с языком разметки SGML, поддержка FTP, и т.д.

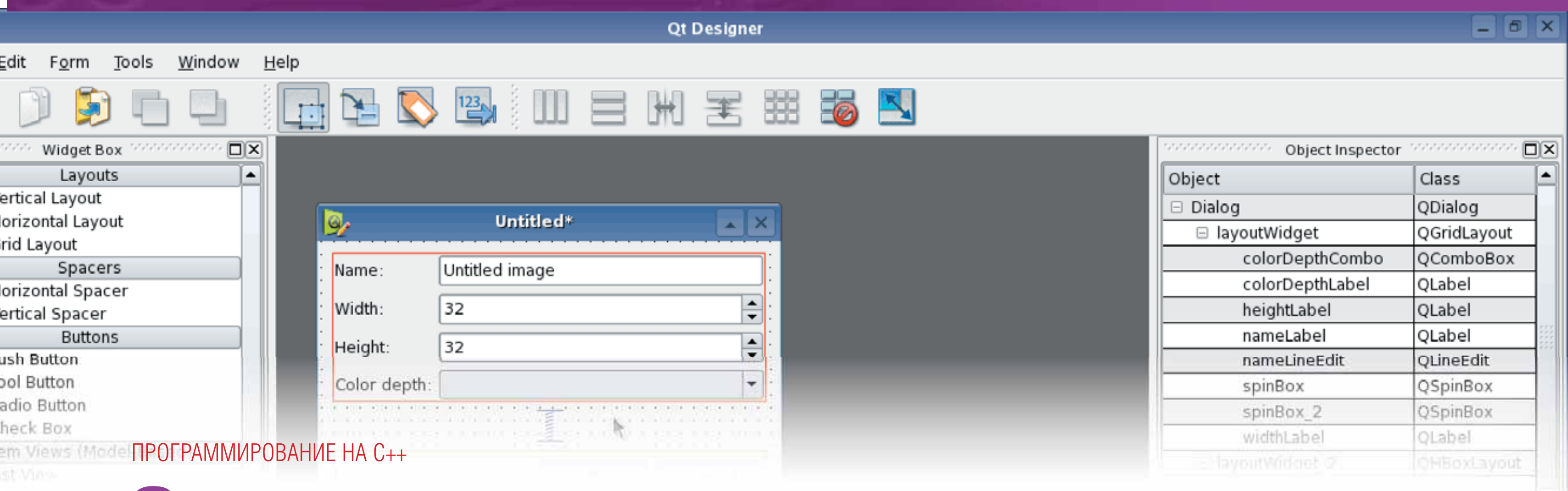
Менеджер почтовых рассылок *Mailman* обеспечивает широкие возможности по управлению списками рассылок, включая web-интерфейс. Интернет-сервер *Medusa* обладает достаточно хорошими характеристиками, позволяя использовать его как для тестовых целей, так и для промышленной эксплуатации. Популярный сервер web-приложений *Zope* также полностью разработан на языке *Python*.

Таким образом, этот язык способен решать весьма серьезные сетевые задачи, причем эти решения, как правило, обладают весьма высокой переносимостью между различными системами и платформами.

В следующий раз мы рассмотрим способы взаимодействия с базами данных, а также убедимся, что *Python* очень хорош и для разработки динамических web-сайтов. LXF

ЧЕРЕЗ МЕСЯЦ

Мы изучим работу с базами данных и посмотрим на Python через призму web-программирования



ПРОГРАММИРОВАНИЕ НА С++

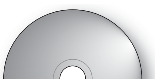
Создаем стандартное KDE-приложение

ЧАСТЬ 5 Сегодня Андрей Боровский расскажет вам, как создаются настоящие KDE-приложения. Ну, или почти настоящие...

Если в чем-то уверен – проверь еще раз.
Девиз параноика.

МЕСЯЦ НАЗАД

На написали и перевели на русский язык простейшее KDE-приложение



НА ДИСКЕ

- Код примеров статьи

В прошлый раз мы научились создавать простейшие KDE-приложения, а также выполнять переводы приложений KDE на разные языки. Было показано, как можно заставить наше приложение использовать ресурсы другого приложения и как добавлять переводы к уже существующим программам. В принципе, изложенных навыков достаточно, чтобы заново выполнить перевод всей среды KDE (сделать, например, перевод с особым цинизмом). Однако, приложения из прошлой статьи не выполняли никакой полезной работы, в том числе потому, что у них отсутствовал сколько-нибудь развитый пользовательский интерфейс. В этой статье мы не будем заниматься переводами (если захотите, можете выполнить их сами), зато напишем «почти настоящее» приложение KDE, использующее меню, панель быстрого доступа и другие интерфейсные элементы, предоставляемые KDE. Наше приложение «почти» (а не совсем) настоящее потому, что оно все еще игнорирует некоторые важные функции среды KDE, однако его уже можно использовать в практических целях. Работающее приложение (исходные тексты вы найдете на диске) выглядит вполне серьезно (рис. 1). Программа позволяет просматривать графические файлы, а также применять к изображениям преобразования: из цветного в черно-белое, изменение контраста и интенсивности (соответствующая функция называется именно так – intensity).

Мы начнем разработку нашего приложения в среде *KDevelop*, используя заготовку **Application framework** (надеюсь, вы помните, как ее найти). Назовем наш проект **images**. В результате выполнения соответствующего мастера будет создана директория **images** с поддиректориями и много файлов **.cpp**, **.h** и других типов. Это обилие может напугать начинающего программиста, но на самом деле все не так страшно. *Application framework* представляет собой заготовку классического офисного приложения KDE. У него есть строка меню, панель быстрого доступа, строка состояния и центральная область, которая используется для ввода/вывода данных (в терминологии «офисного» программирования эта область называется представлением (view). Все эти элементы в нашем приложении уже присутствуют. Самые важные и интересные файлы для нас – **images.h/cpp** и **imagesview.h/cpp**. Эти файлы содержат определения классов *images* и *imagesView* соответственно. Имя первого класса совпадает с именем проекта и приложения, имя второго класса получено



Рисунок 1. Программа "images".

из первого добавлением View. Класс images является потомком класса KMainWindow, реализующего главное окно офисного приложения. Окно KMainWindow позволяет легко управлять такими элементами интерфейса как главное меню, панель инструментов, строка состояния и, конечно, представление. Представление реализовано в классе imagesView. Объект класса images выполняет роль главного визуального элемента для объекта app, класса KApplication. Объект app объявлен в файле **main.cpp**, в котором реализована функция **main()** нашего приложения (нам не нужно модифицировать этот файл). Объекты menuBar, toolBar и statusBar (все они являются членами класса KMainWindow) представляют соответственно строку меню, панель инструментов и строку состояния. Схема взаимодействия классов приложения не должна выглядеть очень сложной (рис. 2, под именами классов подписаны имена объектов этих классов, объявленных в классе images). Класс images выполняет роль связующего звена между элементами пользовательского интерфейса и представлением данных imagesView. Методы этого класса являются обработчиками событий интерфейса и вызывают соответствующие мето-

ЕДИНЫЕ ДЕЙСТВИЯ

Концепция единых действий должна быть хорошо знакома тем, кто программировал в Borland Delphi и C++ Builder. В приложениях с графическим интерфейсом, как правило, одну и ту же команду можно вызвать несколькими способами (из основного меню, с панели инструментов, из контекстного меню). При этом во всех формах вызова команды обычно используются одни и те же элементы – пиктограмма, всплывающая подсказка, и, конечно, функция-обработчик команды. До появления концепции единых действий (actions) программистам приходилось выполнять много повторяющейся работы: для каждого элемента интерфейса нужно было явным образом указывать пиктограмму, подсказку и обработчик команды.

Далее сложности нарастали: если в определенном состоянии программы команду требовалось заблокировать, необходимо было отслеживать состояния всех элементов интерфейса, связанных с этой командой. Как вы уже догадались, концепция единых действий существенно упрощает работу программиста. В рамках этой концепции все общие элементы команды объединяются в один объект (в случае KDE, объект класса *KAction*). Создав объект *KAction* для определенной команды, мы указываем объекту, в каких элементах интерфейса должен присутствовать вызов этой команды. Об остальном объект *KAction* позаботится сам. Если мы свяжем этот объект с одним из меню, будет создана строка меню, если с панелью инструментов – будет создана кнопка и т.п. Если в какой-то момент команда должна быть заблокирована, нам достаточно вызвать метод **setEnabled(FALSE)** для объекта *KAction* для того, чтобы все визуальные представления команды были заблокированы. Аналогично выполняется и разблокирование команды. Можно сказать, что объект *KAction* – это центр управления всеми элементами интерфейса, связанными с соответствующей командой.

ды класса представления.

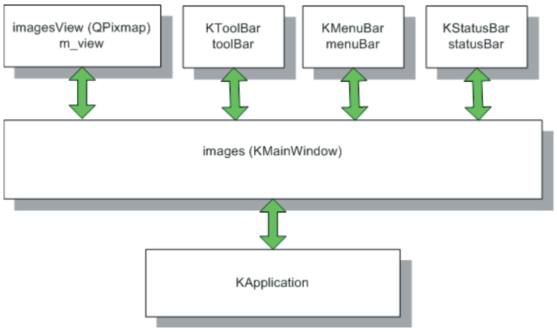


Рисунок 2. Взаимодействие классов приложения.

Для того, чтобы открыть в *KDevelop* файлы исходных текстов, нужно открыть вкладку Группы файлов, расположенную у левого края главного окна среды, и выбрать в ней группу Sources. Перед вами появится список файлов исходных текстов. Откройте нужный файл щелчком мыши.

В файле **imagesview.h** внесите изменения в объявление класса imagesView так, чтобы он стал потомком класса QLabel (напомню, что класс QLabel способен выводить изображения). Класс imagesView является также потомком класса imagesIface, но эта «наследственная линия» нас сейчас не интересует, и мы оставим ее без изменений. Естественно, в файл **imagesview.h** следует добавить директиву

```
#include <qpixmap.h>
```

Удалим из объявления imagesView ненужные члены и добавим нужные. После этого объявление класса будет выглядеть так:

```
class imagesView : public QLabel, public imagesIface
{
   Q_OBJECT
public:
   imagesView(QWidget *parent);
   virtual ~imagesView();
   void toBlacknWhite();
   void setContrast(int c);
   void setIntensity(int i);
};
```

Помимо конструктора и деструктора у класса *imagesView* появились три новых метода. Эти методы выполняют преобразования изображения, хранящегося в imagesView. Метод **toBlacknWhite()** выполняет преобразование палитры из цветной в черно-белую, метод **setContrast()** позволяет настроить контрастность изображения, а с помощью метода **setIntensity()** можно изменить интенсивность.

Теперь перейдем в файл **imagesview.cpp**. Добавим в этот файл две директивы включения заголовочных файлов:

```
#include <kpixmap.h>
#include <kpixmapeffect.h>
```

Теперь приступим к реализации методов:

```
imagesView::imagesView(QWidget *parent)
   : QLabel(parent),
     DCOPObject("imagesIface")
{
   setAlignment(Qt::AlignHCenter|Qt::AlignVCenter);
}

imagesView::~imagesView()
{
}

void imagesView::toBlacknWhite()
{
  if (pixmap() == NULL)
  {
    kdDebug() << k_funcinfo << "Trying to modify an empty pixmap" <<
endl;
    return;
  }
  KPixmap kpm(* pixmap());
  KPixmapEffect::toGray(kpm, FALSE);
  setPixmap(kpm);
}

void imagesView::setContrast(int c)
{
  if (pixmap() == NULL)
  {
    kdDebug() << k_funcinfo << "Trying to modify an empty pixmap" <<
endl;
    return;
  }
   KPixmap kpm(* pixmap());
   KPixmapEffect::contrast(kpm, (c – 50)*5);
   setPixmap(kpm);
}

void imagesView::setIntensity(int i)
{
  if (pixmap() == NULL)
  {
    kdDebug() << k_funcinfo << "Trying to modify an empty pixmap" <<
endl;
    return;
  }
   KPixmap kpm(* pixmap());
   KPixmapEffect::intensity(kpm, (i-50)/10);
   setPixmap(kpm);
```

>>

<<

```
}
```

В конструкторе класса мы устанавливаем горизонтальное и вертикальное выравнивание для содержимого метки.

Для того, чтобы понять, как работают три метода, выполняющих преобразования изображения, рассмотрим классы KDE KPixmap и KPixmapEffect. Первый из этих классов представляет собой улучшенный вариант QPixmap, второй класс предназначен для применения различных эффектов к изображению, хранящемуся в KPixmap. Эффекты реализованы в виде статических методов класса KPixmapEffect. Методы модифицируют содержимое переданного им объекта (а не создают новый). Для выполнения обработки изображения нам приходится преобразовывать QPixmap в KPixmap. Фактически наш класс imagesView представляет собой модифицированный вариант QLabel, способный выполнять преобразования над загруженным изображением. Для вывода сообщения об ошибке (которая никогда не должна возникнуть) мы используем функцию **kdDebug()**, которая возвращает нам объект потока C++ для вывода отладочных сообщений (можно было бы воспользоваться и **printf()**, но это же все-таки C++!). Макрос **k_funcinfo** выводит информацию о функции, из которой был вызван. Функция **kdDebug()** и макрос **k_funcinfo** объявлены в заголовочном файле **kdebug.h**. В том же файле объявлен макрос **k_lineinfo**, который ограничивается вводом информации о файле строке исходного текста, в которой был вызван.

Прежде чем мы перейдем к написанию класса images, ответственного за поведение интерфейса нашей программы, нам понадобится создать один дополнительный элемент интерфейса – диалоговое окно для настройки параметров изображения. Это окно изображено на приведенном выше снимке экрана. Оно содержит горизонтальный ползунок (объект класса *QSlider*) и две кнопки – OK и Cancel. Будет логично, если для создания диалогового окна мы воспользуемся методами визуального программирования, и спроектируем окно в самой среде *KDevelop*, используя встроенный в нее редактор *KDevelop Designer* (*см. врезку*). В окне Automake Manager, щелкните правой кнопкой мыши по строке images (Программа в bin) и выберите команду Создать файл.

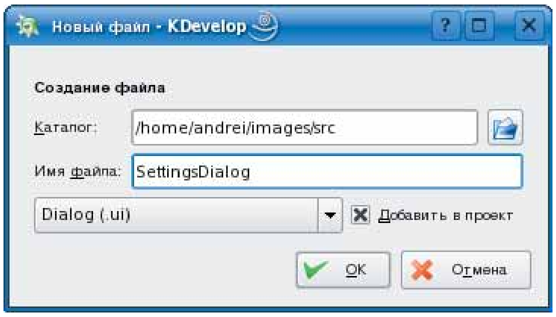


Рисунок 3. Окно создания нового файла.

АЛЬТЕРНАТИВЫ QT DESIGNER

В свое время Linux-разработчиками было начато немало проектов, предназначенных для расширения возможностей визуального программирования *Qt*. Некоторые из этих проектов (их еще можно найти на сайтах *Qt Community*) в идеале должны были приблизиться по удобству и простоте разработке к таким средствам, как *Delphi*. Однако, проверку временем и популярностью выдержали лишь некоторые из них, являющиеся простыми надстройками над *Qt Designer*. В вашем дистрибутиве Linux, скорее всего, присутствуют два таких альтернативных редактора пользовательского интерфейса: *KDevelop Designer* и *Kommander*. *KDevelop Designer* отличается от *Qt Designer* примерно также, как *KPixmap* от *QPixmap*, то есть, практически ничем. Стоит упомянуть также «динамический редактор диалогов» *Kommander*, который создает самостоятельные диалоговые окна и окна мастеров, для использования, например, в сценариях оболочки. В этом редакторе можно не только сконструировать окно со всеми дочерними визуальными элементами, но и проверить, как работают связки сигнал-слот в новом окне.

В открывшемся диалоговом окне (рис. 3) укажите имя нового файла – **SettingsDialog** и тип – **Dialog (ui)**. После этого будет открыт редактор окон, подобный *Qt Designer*, однако для того, чтобы добраться до окна формы, придется «разгрести» множество служебных окон (при разрешении 1024x768 окно формы оказывается полностью «похороненным» под ними). Перенесите в окно формы две кнопки PushButton и ползунок Slider и объедините их в группы (рис. 4). Сигнал **clicked()** кнопки OK соедините со слотом **accept()** класса Form1 (для тех, кто забыл, напомним, что соответствующее окно вызывается командой Соединения... контекстного меню или с помощью клавиши F3). Сигнал **clicked()** кнопки Cancel следует соединить со слотом **reject()** класса Form1. Вызов слотов **accept()** и **reject()** приводит к закрытию диалогового окна. При этом значение, которое окно передает программе после своего закрытия, определяет, какая из кнопок была нажата.

Теперь вернитесь в *Automake Manager*, щелкните правой кнопкой мыши по имени файла **SettingsDialog.ui** и в контекстном меню выберите команду Создать или выбрать класс реализации....

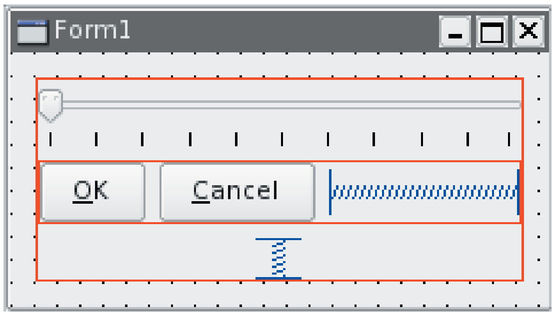


Рисунок 4. Проект диалогового окна.

На самом деле мы создаем не класс реализации, а наследуем от класса формы. В открывшемся окне введите имя нового класса – SettingsDialogImpl. Закройте окна созданных файлов **settingsdialogimpl.h** и **settingsdialogimpl.cpp**. В редакторе форм создайте новый слот (команда Слоты... контекстного меню) **void sliderMoved(int i)** и соедините его с сигналом **sliderMoved(int)** объекта **slider1**. Теперь закройте окно визуального редактора (не забыв сохранить файл **SettingsDialog.ui**) и откройте файл **settingsdialogimpl.h**. В объявление класса SettingsDialogImpl добавьте закрытую переменную **_sliderpos** и метод **getSliderValue()**:

```
class SettingsDialogImpl: public Form1 {
Q_OBJECT
public:
  SettingsDialogImpl(QWidget *parent = 0, const char *name = 0);
public slots:
  int getSliderValue();
protected slots:
  void sliderMoved(int i);
private:
  int _sliderpos;
};
```

В файл **settingsdialogimpl.cpp** добавьте реализацию метода **getSliderValue()** и слота **sliderMoved()**:

```
void SettingsDialogImpl::sliderMoved(int i)
{
  _sliderpos = i;
}
int SettingsDialogImpl::getSliderValue()
{
  return _sliderpos;
}
```

На этом разработка диалогового окна закончена. В файл **images.cpp** необходимо добавить директиву

```
#include "settingsdialogimpl.h"
```

Теперь мы можем приступить к написанию, а точнее, к модификации класса *images*. Дело в том, что этот класс, связующее звено нашего приложения, довольно сложен и значительная его часть была сгенерирована автоматически, так что лучше всего трансформировать класс, созданный по умолчанию, в тот класс, который нам нужен, путем постепенных изменений. Что мы хотим изменить в классе images? Мы хотим удалить все ненужное. Наше приложение предназначено для редактирования существующих графических файлов, но не для создания новых, поэтому удалим метод **fileNew()** Кроме того, дабы не усложнять наш проект сверх меры, мы удалим все, что связано с выводом данных на печать, в том числе метод **filePrint()** и поле **m_printer**. На поле **m_printer** ссылается конструктор images, в него тоже нужно внести изменения. Удалим методы **saveProperties()** и **readProperties()** – они нам сейчас не нужны. Также следует удалить содержимое методов **load()**, **fileSave()** и **fileSaveAs()**, а сами методы – оставить. Если удаление всего ненужного выполнено грамотно, приложение должно скомпилироваться и запуститься. Возможно, вы заметили, что хотя код, выполняющий команды File|New и File|Print удален, пункт меню и кнопка быстрого доступа остались. Для того, чтобы удалить их, нужно перейти в метод **setupActions()** и удалить строки

```
KStdAction::openNew(this, SLOT(fileNew()), actionCollection());
KStdAction::print(this, SLOT(filePrint()), actionCollection());
```

Что делали две удаленные строки? Они добавляли в приложения два стандартных действия (*см. врезку*). Для стандартных действий система сама предоставляет пиктограмму, название и клавишу быстрого доступа. Статическим методам класса KStdAction передается указатель на класс, реализующий обработку команды, слот, обрабатывающий команду, а также указатель на коллекцию действий actionCollection (возвращается методом **actionCollection()**). Программа будет работать даже если слотов, связанных с действиями, не существует, только в стандартный поток вывода будет выведено предупреждающее сообщение. Такова динамическая природа модели сигнал/слот.

Обратите внимание, что мы не прибегаем к визуальному программированию при проектировании главного окна нашего офисного приложения. Дело в том, что для разработки главного окна визуальное программирование нам и не нужно, поскольку вид этого окна однозначно определен принципами построения приложения «офисного» типа.

Теперь подумаем о том, что следует добавить в класс images. В главную строку меню приложения мы добавим пункт Processing, который будет открывать подменю, содержащее команды обработки изображения To Black & White (в черно-белый), Contrast (контраст) и Intensity (интенсивность). Кнопки новых команд нужно добавить на панель быстрого доступа. Все эти изменения должны быть отражены в классе *images*, отвечающем за интерфейс. Ну и конечно, в images нужно добавить код, выполняющий новые команды. Именно в таком порядке мы и будем вносить изменения в класс.

Весь код, отвечающий за добавление новых элементов интерфейса, мы сгруппируем в новом методе **setup_Controls()**. В среде *KDevelop* новые методы классов можно добавить вручную, а можно – автоматически. Откройте вкладку Классы, расположенную с правой стороны главного окна *KDevelop*. Раскройте группу src, щелкните правой кнопкой мыши по классу images и в открывшемся контекстном меню выберите команду Добавить метод.... Откроется окно создания нового метода, похожее на окно создания нового слота в *Qt Designer*.

После добавления нового метода можно приступать к его реализации. В методе **setup_Controls()** мы создаем подменю Processing главного меню, добавляем разделительную полосу на панель быстрого доступа (просто для красоты), создаем и настраиваем три объекта действия (объекты классы KAction), и делаем соответствующие команды доступными в меню и на панели быстрого доступа. Рассмотрим все эти этапы подробнее.

```
void images::setup_Controls()
{
  KPopupMenu  * procMenu =  new  KPopupMenu();  //Processing
  menuBar()->insertItem(i18n("&Processing"), procMenu, 0, 1);
```

Создаем меню Processing. Подменю главного меню должно быть объектом класса KPopupMenu. Для того, чтобы указать, что новое всплывающее меню принадлежит главному меню, мы вызываем метод **insertItem()** поля menuBar (к которому мы получаем доступ с помощью одноименного метода). Кроме указателя на объект класса KPopupMenu методу **insertItem()** передается имя нового меню, его идентификатор id и индекс – позиция в главной строке меню (значению 1 соответствует вторая позиция, после File). Идентификатор должен быть положительным числом. Если **insertItem()** передать отрицательное значение идентификатора, **insertItem()** вернет значение нового уникального идентификатора для меню.

```
  toolBar()->insertLineSeparator(3);
```

Добавляем линию-разделитель

```
  KAction * command = new KAction(i18n("To &Black && White"),
                QIconSet(QPixmap("c1.xpm")), 0, this,
                SLOT(proc_toBlacknWhite()),
                actionCollection(), "bw_command");
```

Создаем объект-действие для команды To Black & White. Файл c1.xpm содержит пиктограмму команды (этот файл должен быть доступен программе во время выполнения). Класс QIconSet преобразует переданный ему графический файл в набор пиктограмм различный размеров и форм (в том числе, в черно-белую пиктограмму для заблокированной команды). Слот **proc_toBlacknWhite()** будет содержать код обработки команды. Последний параметр – имя объекта-команды. Имена могут быть у всех объектов *Qt/KDE*, наследующих QObject, но это первый случай, когда они нам действительно пригодятся.

```
  command->setToolTip(i18n("Transforms image palette to grayscale"));
  command->setEnabled(FALSE);
```

В этих строках мы добавляем всплывающую подсказку и указываем, что по умолчанию команда заблокирована. Последнее нужно для того, чтобы пользователь не пытался обрабатывать еще не существующее изображение. Правда, методы класса imagesView все равно этого не допустят, но лучше перестраховаться (*см. эпиграф*).

```
  command->plug(procMenu);
  command->plug(toolBar(), -1);
```

С помощью метода **plug()** мы указываем объекту действия, что соответствующая команда должна быть доступна в меню procMenu и на панели быстрого доступа.

Добавление двух остальных команд выполняется аналогично. Добавим вызов метода **setup_Controls()** в конструктор images после вызова **setupGUI()**. Теперь можно запустить программу и полюбоваться на новый интерфейс. Правда, слоты для новых команд еще не созданы, так что новые команды пока не работают. Прежде чем создать слоты, вернемся в конструктор images и добавим в него строки

```
actionCollection()->setHighlightingEnabled(TRUE);
connect(actionCollection(),
     SIGNAL(actionStatusText(const QString&)),
     statusBar(), SLOT(message ( const QString& )));
 connect(actionCollection(), SIGNAL(clearStatusText()),
        statusBar(), SLOT(clear()));
```

после вызова **statusBar()->show()**. Это стандартная последовательность вызовов, предназначенная для того, чтобы всплывающие подсказки отображались в строке состояния. Теперь мы можем добавить слот **proc_toBlacknWhite()**, и слоты для двух других команд: **proc_Contrast()** и **proc_Intensity()**. В коде этих слотов (*см. исходные тексты на диске*) для нас нет почти ничего нового, отметим только один момент. Для того, чтобы разблокировать/заблокировать команды пользовательского интерфейса, нужно получить доступ к соответствующему объекту действия. Доступ к коллекции объектов мы получаем с помощью метода **actionCollection()**, а сами объекты можем найти по имени – строке, переданной конструктору Kaction.

Мы научились создавать приложения KDE со стандартным графическим интерфейсом. Но в мире программирования KDE это – только начало. В следующих статьях мы узнаем, как создавать KDE-приложения специальных типов, а также познакомимся с некоторыми полезными службами KDE, такими как менеджер сессий. LXF

ЧЕРЕЗ МЕСЯЦ

Мы научимся создавать специализированные KDE-приложения и познакомимся со службами KDE.



ПРОГРАММИРОВАНИЕ ДЛЯ UNIX

Очереди сообщений и семафоры

ЧАСТЬ 3 Каналы – отнюдь не единственное средство межпроцессного взаимодействия в Unix. В этом выпуске Андрей Боровский расскажет еще о нескольких механизмах IPC: очередях сообщений, разделяемой памяти и семафорах.

Я должен отметить, что основная цель компьютерных наук, – устранение неразберихи – так и не была выполнена

Эсгар Дейкстра

МЕСЯЦ НАЗАД

Мы начали рассмотрение механизмов IPC с каналов

НА ДИСКЕ

- Код примеров статьи

Мы продолжаем изучение механизмов взаимодействия между процессами в Linux. Каналы различных типов, рассмотренные в предыдущей статье, существовали в Unix практически с самого начала. Позже к ним были добавлены и другие механизмы межпроцессного взаимодействия. Мы остановимся на трех механизмах, которые появились в Unix System V и были описаны в System V Interface Definition (SVID). В настоящее время эти механизмы поддерживаются почти всеми Unix-системами (очереди сообщений не поддерживаются в Mac OS X 10.3 – *см. список литературы 1*). Интерфейсы трех механизмов SVID IPC подобны. Чтобы разные процессы могли получить доступ к одному объекту системы, они должны «договориться» об идентификации этого объекта. Роль идентификатора для всех объектов System V IPC выполняет ключ – уникальное число-идентификатор объекта. Чтобы использовать один и тот же объект, программы должны использовать один и тот же ключ. Для каждого объекта IPC предусмотрены специальные функции чтения и записи, а также управляющая функция.

Сообщения

Механизм сообщений Linux похож на механизм сообщений, используемый в графических многооконных средах. Сообщения накапливаются в очередях и могут изыматься из очередей последовательно или в произвольном порядке. Каждая группа процессов может создать одну или несколько очередей для обмена сообщениями, а одна очередь сообщений может использоваться совместно более чем двумя процессами. Сообщение определяется как «последовательность байтов, передаваемая от одного процесса другому». Система сообщений SVID обладает следующими свойствами:

- Возможность накопления сообщений в очереди. Приложения, использующие сообщения для обмена данными, создают свою собственную очередь сообщений, которая может (и должна) быть удалена приложением-владельцем в момент завершения его работы.
- Возможность произвольного выбора сообщений из очереди на основе назначенных им идентификаторов. Эта возможность позволяет организовать приоритетную обработку сообщений, а также идентифицировать сообщения, посылаемые разными приложениями, участвующими в обмене данными.
- Произвольная структура и размер сообщения.

Последний пункт требует уточнения. Максимальный размер сообщения и максимальное количество сообщений в очереди ограничены, причем не существует единого для всех Unix-систем способа определить эти ограничения. В Linux максимальная длина сообщения в байтах задана константой MSGMAX, определенной в фале **<linux/msg.h>**, а максимальное число сообщений – константой MSGMNG из того же файла. На платформе IA32 размер сообщения не может превышать 8

килобайт, а длина очереди – 16384 (16K) сообщений. Структура данных, использующаяся для передачи сообщений, может быть определена следующим образом:

```
struct msgp
{
 long mtype;
 ... // Любые другие поля
};
```

Поле mtype является единственным обязательным полем в приведенной структуре. В этом поле хранится произвольный идентификатор сообщения, который может интерпретироваться как тип передаваемых данных. Кроме поля **mtype**, структура данных сообщения может содержать любое количество других полей любых типов.

В качестве примера использования очередей рассмотрим совместную работу двух программ – клиента и сервера. Исходные тексты вы найдете на диске (файлы **msgcli.c** и **msgserv.c** соответственно). Чтобы программы могли обмениваться сообщениями, они должны использовать один и тот же формат сообщений и идентификатор очереди. Эти данные, общие для клиента и сервера, удобно вынести в отдельный заголовочный файл (мы назовем его **msgtypes.h**). Наша структура данных выглядит так:

```
#define MAXLEN 512
struct msg_t
{
 long mtype;
 int snd_pid;
 char body[MAXLEN];
};
```

Помимо поля **mtype**, мы вводим поле **snd_pid**, которое будет содержать идентификатор процесса-отправителя сообщения, и поле **body**, которое предназначено для текста сообщения. Мы могли бы определить несколько разных структур для сообщений разных типов. Значение поля **mtype** указывало бы, с какой структурой мы имеем дело. Поле mtype может быть не только идентификатором типа сообщения. С его помощью можно указать, например, приоритет сообщения. Используя функцию произвольной выборки сообщений, приложение может считывать в первую очередь сообщения с более высоким приоритетом.

Кроме структуры сообщения нам следует определить ключ очереди. Для получения уникального ключа можно использовать функцию **ftok(3)**, однако руководство по работе с функциями SVID рекомендует выбирать значения самостоятельно, поэтому в нашем примере мы определим ключ как константу в файле **msgtypes.h**:

```
#define KEY 1174
```

Маловероятно, что в системе уже существует другая очередь сообщений с ключом 1174. В принципе, программа, создающая объект IPC, может узнать, существует ли уже такой объект (*см. ниже использование флага IPC_EXCL*), однако толку от этого не много. Допустим, процесс установил, что объект с указанным идентификатором существует, но что ему делать? Процесс может выбрать другой идентификатор из какого-нибудь пула, однако о новом идентификаторе нужно как-то оповестить другие процессы. Для оповещения можно использовать именованные каналы, для которых, в свою очередь, необходим уникальный идентификатор... Уникальность идентификатора файловых каналов основана на уникальности имен файловой системы (имеются в виду полные имена, начиная с корня). Функция **ftok()**, которую мы рассмотрим ниже, тоже пытается генерировать идентификаторы, основываясь на уникальности имен файловой системы. Кроме того, проверка существования объекта IPC может «обмануть» процесс, если существующий объект был создан предыдущим экземпляром того же процесса, выгруженным из системы в результате серьезной ошибки.

Рассмотрим теперь работу сервера. Сервер получает сообщение, переданное клиентом, распечатывает сообщение на экране терминала, возвращает клиенту сообщение "OK!", ждет подтверждения, что клиент получил ответ, затем удаляет очередь и завершает работу. Программу-сервер следует запустить до запуска программы-клиента.

Текст программы (**msgserv.c**), как всегда, начинается с заголовочных файлов. Все типы, константы и функции, использующиеся при работе с сообщениями, становятся доступны при включении в текст программы файлов **<sys/ipc.h>** и **<sys/msg.h>**. Очередь сообщений создается при помощи функции **msgget(2)**:

```
msgid = msgget(KEY, 0666 | IPC_CREAT);
```

Первый параметр **msgget()** – ключ, гарантирующий уникальность очереди. Он числовой, поэтому его можно спутать с другим числом – идентификатором очереди, который присваивает система. Помните, что ключ нужен только для открытия очереди, а для работы с ней используется идентификатор. Второй параметр представляет собой комбинацию маски прав доступа к создаваемой очереди (аналогичной маске прав доступа к именованным каналам) и нескольких дополнительных флагов: кажется, программист, писавший функции SVID IPC, сильно экономил на переменных-параметрах. Флаг **IPC_CREATE** указывает, что в результате вызова **msgget()** должна быть создана новая очередь. При установке флага **IPC_EXCL**, функция **msgget()** вернет сообщение об ошибке, если очередь с указанным ключом уже существует. В случае успеха **msgget()** возвращает положительное значение – идентификатор созданной очереди.

Передача и получение сообщений выполняется при помощи функций **msgsnd(2)** и **msgrcv(2)** соответственно. Первым параметром обеих функций является идентификатор очереди, возвращенный функцией **msgget()**. Во втором параметре передается размер структуры сообщения. Как было сказано выше, программа, читающая сообщения из очереди, должна указать размер сообщения, соответствующий ожидаемому идентификатору, и может читать сообщения разного размера в ситуации, когда программа ждет сообщений определенного типа. На диске есть пример **polymsgserv/polymsgcli**, демонстрирующий этот подход. Третьим параметром функции **msgrcv()** является идентификатор сообщения. Если значение этого параметра больше нуля, из очереди будет извлечено сообщение с соответствующим значением поля **mtype**. Если этот параметр равен нулю, из очереди будет извлечено первое по порядку сообщение, а если параметр отрицательный, из очереди будет извлечено первое сообщение, чей идентификатор меньше либо равен абсолютному значению параметра. Последний параметр в функциях **msgsnd()** и **msgrcv()** позволяет задать дополнительные флаги. Обычно функция, читающая сообщение из очереди, приостанавливает выполнение программы до тех пор, пока извлечение сообщения не будет выполнено, то есть пока в очереди не появится сообщение ожидаемого типа. Именно так работает эта функция в наших программах. При указании флага **IPC_NOWAIT**, **msgrcv()** вернет сообщение об ошибке, если на момент вызова в очереди отсутствует подходящее сообщение.

В нашем примере сервер и клиент используют разные идентификаторы для посылаемых сообщений. Это сделано для того, чтобы программа, вызывающая последовательно **msgsnd()** и **msgrcv()**, не извлекала из очереди свои собственные сообщения. Наш сервер записывает в очередь сообщения со значением mtype, равным 1, а считывает – со значением, равным 2 (у программы-клиента все наоборот).

Для удаления очереди используется функция **msgctl(2)**, которая, как и все функции ***ctl()**, может выполнять множество разных действий (например, получение данных о состоянии очереди). Первый параметр этой функции, как всегда, идентификатор очереди, второй параметр – команда (**IPC_STAT**, **IPC_SET** или **IPC_RMID**). Третий параметр используется в вызовах-запросах (то есть, когда второй параметр равен **IPC_STAT**), а также для конфигурации очереди (команда **IPC_SET**). В нем передается указатель на структуру **msgid_ds**, поля которой содержат значения различных параметров очереди. Функция возвращает статус выполнения команды. Вызов

```
msgctl(msgid, IPC_RMID, 0);
```

удаляет очередь с идентификатором **msgid**.

Рассмотрим теперь программу-клиент. Первым делом она должна получить идентификатор очереди. Для этого используется функция **msgget()** с тем же ключом очереди, что и у сервера, с маской прав доступа, но без дополнительных флагов. Она возвращает или идентификатор уже существующей очереди с данным ключом или **-1**, если очередь не существует:

>>

НЕАТОМАРНОСТЬ СПИН-БЛОКИРОВОК

Простейший алгоритм разделения доступа (который сразу приходит в голову) можно описать схематически с помощью следующей конструкции:

```
while (spin_lock == TRUE);
spin_lock = TRUE;
... // Доступ к разделяемому ресурсу
spin_lock = FALSE;
```

Проблема заключается в том, что этот алгоритм простых спин-блокировок (spin locks) не только прожорлив, но и не гарантирует надежного разграничения доступа. Допустим, что один процесс начал проверять значение переменной **spin_lock**, дождался того момента, когда оно станет равно **FALSE**, и переходит к строке

```
spin_lock = TRUE;
```

В многозадачной системе (особенно на нескольких процессорах) процесс, ожидающий доступа к разделяемому ресурсу, может «вклиниться» в тот момент, когда другой процесс уже проверил, но еще не изменил значение **spin_lock**. Второй процесс проверит значение, все еще равное **FALSE**, и оба процесса окажутся в критической области и получат доступ к разделяемому ресурсу. Описанная проблема вызвана тем, что операция «проверить значение – изменить значение» неатомарна, то есть ее выполнение может быть прервано другим процессом. Первый алгоритм, гарантирующий синхронизацию при использовании неатомарных операций, придумал Т. Деккер, а применил Э. Дейкстра в 1965 году. В 1981 году Г.Петерсон предложил более простой алгоритм.



<<

```
msgid = msgget(KEY, 0666);
if (msgid == -1)
{
 printf("Server is not running!\n", msgid);
 return EXIT_FAILURE;
}
```

Далее клиент считывает строку, вводимую пользователем, формирует сообщение, записывая в поле **mtype** значение **2**, отправляет сообщение и ждет ответ сервера – сообщения с идентификатором **1**. Скомпилируйте обе программы (можете просто скомандовать **make msgdemo**), запустите сначала сервер, потом, в другом окне терминала, клиент. Напечатайте в окне клиента строку и нажмите ввод.

Разделяемая память

Спецификация SVID описывает интерфейс для работы с разделяемыми блоками памяти. Разделяемые блоки памяти представляют собой область памяти, отображенную адресное пространство нескольких процессов. Если один процесс запишет данные в разделяемую область, другой процесс может считывать их оттуда как из собственной области глобальной памяти. Мы продемонстрируем использование разделяемой памяти на уже знакомом примере клиент-сервер. Как и в случае с сообщениями, нам нужно описать общие структуры данных для клиента и сервера в заголовочном файле (на диске это файл **shmemtypes.h**):

```
#define FTOK_FILE "./shmemserv"
#define MAXLEN 512
struct memory_block
{
 int server_lock;
 int client_lock;
 int turn;
 int readlast;
 char string[MAXLEN];
};
```

Механизм разделяемой памяти не налагает никаких ограничений на структуру блока памяти. Структура *memory_block* определенна нами, исходя исключительно из наших собственных потребностей. Первые четыре поля структуры *memory_block* – служебные, они нужны для реализации модифицированного алгоритма Петерсона, о котором будет сказано ниже. Последнее поле предназначено собственно для передачи данных. В нашем заголовочном файле мы не определяем ключ для идентификации разделяемого блока, но указываем имя некоего файла (в нашем случае – исполнимого файла сервера). Это имя будет передано функции **ftok()** для получения ключа. Естественно, это метод сработает только если сервер будет скомпилирован под именем **shmemserv**. Рассмотрим исходный код инициализации сервера:

```
key_t key;
int shmid;
struct memory_block * mblock;
key = ftok(FTOK_FILE, 1);
shmid = shmget(key, sizeof(struct memory_block), 0666 | IPC_CREAT);
mblock = (struct memory_block *) shmat(shmid, 0, 0);
```

Как уже отмечалось, **ftok()** генерирует ключ, используя в качестве «затравки» имя файла, в нашем случае – имя исполнимого файла сервера. Использование имени самой программы для генерации ключа до некоторой степени гарантирует уникальность ключа. Функции для работы с разделяемой памятью объявлены в файлах **sys/ipc.h sys/shm.h**. Разделяемый блок памяти выделяется при помощи функции **shmget(2)**, которой передаются три параметра. В первом параметре передается ключ, идентифицирующий выделяемый блок памяти. Второй параметр позволяет указать размер блока в байтах. В третьем параметре передается маска прав доступа и флаги, аналогичные флагам **msgget()**. Функция **shmget()** возвращает идентификатор выделенного блока памяти (его не следует путать с указателем на блок). Чтобы получить указатель на созданный блок разделяемой памяти, этот блок нужно отобразить в локальное адресное пространство процесса. Отображение блока разделяемой памяти в адресное пространство процесса выполняет функция **shmat(2)**. У этой функции тоже три параметра. Первый параметр – идентификатор, возвращенный функцией **shmget()**. Во втором параметре передается желательный начальный адрес для отображения разделяемого блока в локальном адресном пространстве. Функция **shmat()** «постарается» отобразить разделяемый блок в локальное пространство, начиная с указанного адреса, но успешный результат не гарантирован. Если во втором параметре **shmat()** передать нулевое значение, функция сама выберет начальный адрес области отображения. Значение желательного адреса должно быть выравнено по границе страничных областей. Можно также не выравнивать адрес, но передать в третьем параметре функции флаг **SHM_RND**, и тогда функция сама скорректирует значение адреса. Среди дополнительных флагов, которые можно передать в третьем параметре, отметим флаг **SHM_RDONLY**, который присваивает отображаемой области статус «только для чтения». При успешном выполнении функция **shmat()** возвращает указатель на начало области отображения, с которым мы можем работать как с обычным указателем на выделенный блок памяти. Чтобы понять дальнейшую работу сервера, следует иметь в виду, что сами по себе объекты разделяемой памяти не предоставляют никаких средств синхронизации доступа, так что нам приходится самим позаботиться об этих средствах. Для синхронизации работы клиента и сервера и разграничения доступа мы используем упомянутый уже алгоритм Петерсона (*См. список литературы, 2*), который позволяет разграничить доступ к блоку разделяемой памяти, используя неатомарные операции.

Функция **shmdt(2)** удаляет область отображения в локальном адресном пространстве (но не удаляет блок разделяемой памяти). Блок разделяемой памяти удаляется вызовом

```
shmctl(shmid, IPC_RMID, 0);
```

который и по форме, и по сути подобен приведенному выше вызову **msgctl()**. Клиент получает доступ к блоку разделяемой памяти с помощью вызовов

```
shmid = shmget(key, sizeof(struct memory_block), 0666);
if (shmid == -1)
{
 printf("Server is not running!\n");
 return EXIT_FAILURE;
}
mblock = (struct memory_block *) shmat(shmid, 0, 0);
```

При этом мы проверяем, запущен ли сервер. Далее клиент читает информацию, записанную в разделяемый блок сервером, считывает строку с терминала и передает строку серверу, используя механизм спин-блокировок. Скомпилируйте обе программы (скомандовав **make shmemdemo**), запустите сервер, затем клиент и набирайте строки в окне клиента. Работа программ завершится, когда вы наберете **q** и нажмете ввод. Поработав с программами, вы, конечно, обратили внимание на медлительность, с которой сервер отвечает клиенту. Кроме

того, вы могли заметить существенный рост потребления ресурсов процессора при работе программ. Виной всему спин-блокировки, которые используются в алгоритме Петерсона. Именно из-за спин-блокировок процессор проводит значительную часть времени в цикле непрерывного опроса значения переменной. Сам алгоритм Петерсона сегодня можно найти только в учебниках по разработке операционных систем, хотя и современные ОС его практически не используют. Вместо этого ОС создают объекты синхронизации, контролирующие доступ к критическим секциям с помощью специальных атомарных операций процессора, и предоставляют пользовательским программам доступ к этим объектам. Именно такими объектами являются рассмотренные далее семафоры.

Семафоры

Семафоры широко используются как средство синхронизации потоков и процессов. В Unix-системах реализованы три типа семафоров – семафоры System V, семафоры POSIX и семафоры в разделяемой памяти. Поскольку статья посвящена System V IPC, мы рассмотрим семафоры System V. Подробное описание всех трех типов семафоров можно найти в (*См. список литературы, 3*).

Состояние семафора определяется значением некоторой внутренней переменной и переданным ему параметром. В зависимости от этих значений семафор либо приостанавливает выполнение обратившегося к нему потока (до тех пор, пока другой поток не переведет семафор в другое состояние), либо изменяет значение внутренней переменной, разрешив потоку дальнейшее выполнение. Следующая функция иллюстрирует логику работы семафора в зависимости от значения переменной состояния (semvalue) и управляющей переменной **sem_op**.

```
void semaphore (int sem_op)
{
 static int semvalue; // Внутренняя переменная
 if (sem_op != 0)
 {
 if (sem_op < 0) while (semvalue < ABS(sem_op));
 semvalue += sem_op;
 }
 else while (semvalue != 0);
}
```

Отрицательное значение sem_op соответствует операции проверки доступности ресурса и вызывает приостановку потока, если доступ к ресурсу заблокирован. Положительное значение сигнализирует о высвобождении ресурса. Приведенная выше функция **semaphore()** описывает поведение многозначного семафора (general semaphore). Именно такие семафоры используются в System V IPC.

Перепишем клиент и сервер из предыдущего примера, заменив спин-блокировки семафорами (на диске вы найдете исходный текст сервера в файле **semserv.c**, а исходный текст клиента – в файле **semcli.c**. Все, что касается семафоров, определено в файле **<sys/sem.h>**. Программа-сервер создает семафоры с помощью вызова

```
semid = semget(key, 2, 0666|IPC_CREAT);
```

Функция **semget(2)** похожа на **msgget()** и **shmget()**, но у нее есть дополнительный параметр – количество создаваемых семафоров. Дело в том, что многим процессам, использующим семафоры, требуется более одного семафора (тогда как блоки разделяемой памяти и очереди сообщений обычно существуют в единственном экземпляре). Определение уникального ключа для каждого из нескольких семафоров затруднительно, поэтому функция **semget()** позволяет несколько семафоров сразу. Нашему приложению понадобится два семафора. Нам понадобилось бы три семафора, если бы... Впрочем, это уже совсем другая история. Первый семафор указывает, должен ли сервер читать запись, сделанную клиентом, второй – должен ли клиент читать запись, сделанную сервером. Таким образом, мы приказываем **semget()** создать сразу два семафора. В случае успешного завершения функция **semget()** возвращает идентификатор нового массива семафоров.

Для определения состояния семафора используется структура **sembuf**. В ней определено много полей (конкретный набор зависит от

РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА:

1. W. R. Stevens, S. A. Rago, **Advanced Programming in the UNIX® Environment: Second Edition**, Addison Wesley Professional, 2005
2. Таненбаум Э. С., Вудхалл А. С., **Операционные системы: разработка и реализация**. – СПб.: Питер, 2005
3. Стивенс У., **UNIX: Взаимодействие процессов**. – СПб.: Питер, 2003

версии Unix-системы), из которых обязательными являются три:

- **short sem_num** – номер семафора (в массиве), над которым выполняется операция (нумерация начинается с нуля).
- **short sem_op** – число, изменяющее состояние семафора.
- **short sem_flg** – дополнительные флаги.

Как и в приведенном выше схематическом примере работы семафора, отрицательное значение **sem_op** соответствует операции проверки доступности ресурса и вызывает приостановку потока, если ресурс недоступен. Положительное значение заставляет семафор высвободить ресурс (или приблизиться к этому). Указатель на массив структур **sembuf** (по структуре на семафор) передается как второй параметр функции **semop(2)**, которая либо изменяет состояние семафора, либо приостанавливает вызывавший поток. Первый параметр этой функции – идентификатор, возвращенный **semget()**. В третьем параметре передается число записей в массиве **sembuf**. Вот как, например, мы указываем, что клиент может записывать данные в разделяемую область:

```
buf[1].sem_op = 1;
semop(semid, (struct sembuf*) &buf[1], 1);
```

А эти строки приостановят сервер, пока клиент не изменит значение первого семафора:

```
buf[0].sem_op = -1;
semop(semid, (struct sembuf*) &buf, 1);
```

Получая разрешение на доступ к разделяемой области, процесс производит чтение/запись, разрешает доступ другому процессу, запрещает доступ себе и приостанавливается. Удаление семафора выполняет с помощью функции **semctl()**, в которой, кроме прочего, нужно указывать число семафоров:

```
semctl(semid, 2, IPC_RMID);
```

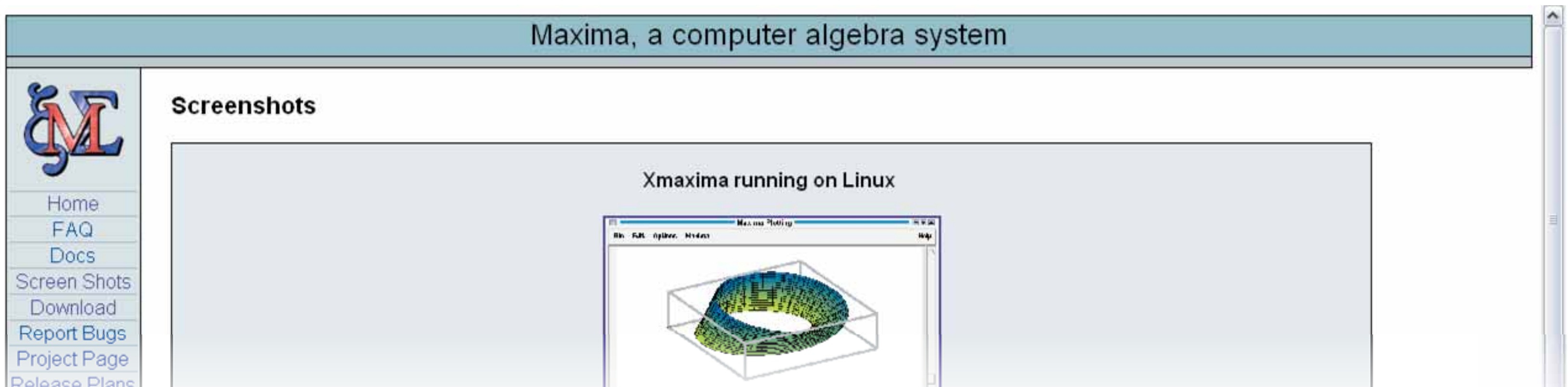
Скомпилируйте сервер под именем **semserv**, а клиент под именем **semcli**, (или командуйте **make semdemo**) запустите клиент и сервер. Вы увидите, что обмен данными выполняется гораздо быстрее, а процессор загружается гораздо меньше, чем в случае использования спин-блокировок.

При всем богатстве выбора средств взаимодействия между процессами в Unix/Linux, самыми популярными средствами были и остаются сокеты. Ими мы и займемся в следующий раз. LXF



ЧЕРЕЗ МЕСЯЦ

Мы окунемся в мир сетевого программирования и познакомимся с сокетами – базовыми «кирпичиками», из которых строятся и мощные сервера, и легкие клиентские приложения.



СИМВОЛЬНЫЕ ВЫЧИСЛЕНИЯ

Maxima – функции и операторы

ЧАСТЬ 2 Понять философию сложного пакета – уже половина дела, однако, для того, чтобы уметь читать, надо хотя бы освоить азбуку. Сегодня Тихон Тарнавский расскажет вам об операторах (или функциях?) Maxima.

МЕСЯЦ НАЗАД

Мы познакомились с идеологией Maxima и освоили простейшие операции

Операторы Максимы

Продолжаю знакомить вас с возможностями свободной программы символьных вычислений *Maxima*. Начну в этот раз с краткого рассказа об основных операторах *Maxima* и некоторых их свойствах.

На самом деле в Максиме нет чёткого разграничения между операторами и функциями. Более того, каждый оператор – это на самом деле функция:

$$\text{(\%i1)}\quad "+"(1,2,3)$$

$$\text{(\%o1)}\quad 6$$

$$\text{(\%i2)}\quad "*"("+"(a,b),"/"(c,d))$$

$$\text{(\%o2)}\quad \frac{(b+a)\,c}{d}$$

Здесь имена функций-операторов берутся в кавычки лишь потому, что содержат символы, нестандартные для имён функций. Это похоже на работу в командной оболочке Unix, где, если в имя файла входят управляющие символы, вы можете либо взять это имя в кавычки, либо экранировать каждый такой символ обратным слэшем. В *Maxima* допустимы те же два варианта: например, вместо "+" можно было бы написать \+.

Итак, все встроенные операторы максимы являются функциями; более того, вы можете наделить любую (в том числе свою собственную) функцию определёнными свойствами, которые фактически превращают её в оператор. Подробнее об этом я расскажу в следующих выпусках.

Таким образом, разделение на функции и операторы в *Maxima* достаточно условно. Посему в этом разделе речь пойдёт не только о некоторых операторах, но и о нескольких функциях, которые по природе своих действий сходны с операторами. Наиболее привычные операторы уже упоминались в предыдущей статье: **+**, **–**, *****, **/**, **^** или ****** (возведение в степень) и функцию **sqrt(x)** (квадратный корень). Сегодня мы поговорим ещё о нескольких достаточно распространённых.

Точкой обозначается матричное произведение. В документации утверждается, что сама точка при этом должна быть отделена пробелами от обоих своих операндов – дабы не спутать её с точкой десятичной. На самом деле мне не удалось добиться от Максимы неадекватной реакции и в «беспробельном» варианте; что и логично, так как всё равно эти две разные ипостаси точки можно различить по контексту: ведь цифры именами матриц быть не могут. Так что, думаю, можете смело писать и без пробелов.

$$\text{(\%i1)}\quad \mathrm{matrix}([1,2,3],[0,a,0],[0,0,b])$$

$$\text{(\%o1)}\quad \begin{pmatrix} 1 & 2 & 3 \\ 0 & a & 0 \\ 0 & 0 & b \end{pmatrix}$$

$$\text{(\%i2)}\quad \mathrm{matrix}([1,2],[0,a],[0,b])$$

$$\text{(\%o2)}\quad \begin{pmatrix} 1 & 2 \\ 0 & a \\ 0 & b \end{pmatrix}$$

$$\text{(\%i3)}\quad \%o1.\%$$

$$\text{(\%o3)}\quad \begin{pmatrix} 1 & 3\,b+2\,a+2 \\ 0 & a^2 \\ 0 & b^2 \end{pmatrix}$$

В случае, если заданные матрицы не могут быть перемножены из-за несовпадающих размерностей, *Maxima* выдаст сообщение об ошибке:

$$\text{(\%i4)}\quad \%o2.\%o1$$

```
incompatible dimensions - cannot multiply
 -- an error.  Quitting.  To debug this try debugmode(true);
```

Восклицательный знак, стоящий после своего аргумента (т.е. постфиксный оператор), традиционно обозначает факториал. Не менее традиционно, двумя восклицательными знаками обозначен полуфакториал [произведение всех четных (для четного операнда) или нечетных чисел, меньших либо равных данному, – прим. ред.]. Функции **abs(x)** и **signum(x)** возвращают, как опять же нетрудно догадаться, модуль и знак числа. А функции **max(x1,…,xn)** и **min(x1,…,xn)** – соответственно максимальное и минимальное из заданных чисел.

Тут стоит остановиться на нескольких моментах. Во-первых, все функции и операторы *Maxima* работают не только с действительными, но и комплексными числами. Сами комплексные числа записываются в Максиме в алгебраической форме, с мнимой единицей, обозначенной

через **%i**; то есть в виде **a+b*%i**, где **a** и **b** – соответственно действительная и мнимая части числа.

Так, факториал задан в наиболее общем виде и представляет собой, по сути, гамма-функцию (точнее, **x! = gamma(x+1)**), то есть определён на множестве всех комплексных чисел, кроме отрицательных целых. При этом факториал от натурального числа (и нуля) автоматически упрощается до натурального же числа:

(%i1) $0!; 6!; 28!;$

(%o1) 1
(%o2) 720
(%o3) $304888344611713860501504000000$

Точно так же и модуль определён для всех комплексных чисел (напомню, что |a+b*i|=sqrt(a^2+b^2)). Минимум, максимум и знак определены, естественным образом, только для действительных чисел, так как комплексные числа общего вида, как известно, между собой несравнимы.

Второй важный момент: когда некоторая встроенная функция или оператор *Maxima* не может получить для переданного выражения однозначный результат (ввиду недостаточности данных) – она пытается максимально упростить это выражение. (Для некоторых функций такое автоупрощение регулируется специальными параметрами.) Например, если x не задан:

(%i1) $\mathrm{abs}(-x)$

(%o1) $|x|$

(%i2) $\mathrm{signum}(-x)$

(%o2) $-\mathrm{signum}(x)$

(%i3) $\mathrm{put}(\mathrm{trylevel}, 3, \mathrm{maxmin})\$$

(%i4) $\max(x, -x)$

(%o4) $|x|$

(%i5) $\max(x, x+1, x-\mathrm{abs}(a))$

(%o5) $x+1$

(%i6) $\max(x, 2x, 3x)$

(%o6) $\max(3x, x)$

Подобные упрощения, равно как и «раскрытие» факториалов и арифметических операторов, не считаются вычислениями, а следовательно оператор блокировки вычислений их не предотвращает:

(%i1) $'(8!-4!\,3!)$

(%o1) 40176

(%i2) $x: 2\$\ '(\mathrm{signum}(-x))$

(%o3) $-\mathrm{signum}(x)$

Как вы, вероятно, помните, в прошлый раз кроме упомянутого только что оператора блокировки вычислений мы познакомились с оператором присвоения значений, или, иначе, именования выражений, **:** . В *Maxima* существуют и другие операторы именования, из которых нам на данный момент интересен один – оператор задания функции. Обозначается он через **:=**, и аналогии здесь прослеживаются не с языками *Pascal* или *Algol*, как может показаться на первый взгляд, а с другими обозначениями самой Максимы: с одной стороны определение функции можно воспринимать как уравнение (которое обозначается знаком **=**), а с другой – оно родственно назначению имени некоторому выражению (то есть **:**). То есть определение функции можно в какой-то мере считать симбиозом этих двух выражений – и оттого вполне логично, что оно обозначается обоими их символами. (В продолжение этой аналогии могу добавить, что в *Maxima* есть и расширенные варианты операторов присвоения и назначения функции, обозначаемые соответственно через **::** и **::=**.)

(%i1) $f(x,y) := x\sin(y) + \frac{\cos(y)}{x}\$$

(%i2) $f(1,y); f(x,0)$

(%o2) $\sin y + \cos y$

(%o3) $\frac{1}{x}$

(%i4) $\mathrm{integrate}(f(x,y),x); \mathrm{integrate}(f(x,y),y)$

(%o4) $\frac{x^2\sin y}{2} + \log(x)\cos y$

(%o5) $\frac{\sin y}{x} - x\cos y$

Думаю, основы работы с функциями самоочевидны по аналогии с приведенным примером, а подробнее об этом мы поговорим в следующих выпусках.

Функция вычисления всего

А сейчас я расскажу о том, что было обещано в прошлый раз: о возможностях управлять процессом вычислений вводимых вами выражений. В прошлый раз, о чём я уже вспоминал, было упомянуто только одно такое

>>

О РАБОТЕ В МАТЕМАТИЧЕСКОМ РЕЖИМЕ ВВОДА РЕДАКТОРА TEXMACS

Первое слово – про апостроф, который используется в *Maxima* для блокировки вычислений. В математическом режиме привычной клавишей вводится несколько другой «апостроф», обозначающий производную. Поэтому для ввода апострофа, блокирующего вычисления, нужно внутри математического режима ввода создать поле текстового ввода – и уже в нём ввести обычный текстовый апостроф. По умолчанию это делается комбинацией клавиш A-$, что в зависимости от настроек *TeXmacs* может расшифровываться как Alt+Shift+4 или Win+Shift+4. После ввода апострофа можно с помощью стрелки влево выйти из поля текстового ввода и продолжать пользоваться всеми прелестями ввода математического.

И второе слово – насчёт ввода различных символов, к которым в математическом режиме либо в самом *TeXmacs* привязаны некоторые клавиатурные сокращения, т.е. при простом нажатии на клавишу обычные символы не вводятся, а вместо этого происходит некое другое привязанное к этой клавише действие. Среди таких переназначенных символов, к примеру, – «$» и «\». Для того, чтобы отменить специальное действие и вместо него просто ввести обозначенный на клавише символ, нужно непосредственно перед этой клавишей нажать Shift+F5.

То же самое можно сказать и про кавычку, к которой уже глобально в *TeXmacs* привязан по умолчанию ввод «фигурных» кавычек. Здесь есть два варианта: либо предварить ввод кавычки нажатием той же самой комбинации Shift+F5; либо поменять умолчательное поведение редактора с помощью пункта меню Редактировать –> Предпочтения –> Клавиатура –> Автоматические кавычки –> Никаких – правда, тогда перед вводом кавычки придётся, так же как и для апострофа, переходить внутри математического режима в текстовый.

ИНТЕРФЕЙСЫ К MAXIMA

Кроме двух описанных в прошлый раз интерфейсов к Максиме – *wxMaxima* и TeXmacs, – есть, как уже говорилось, и другие варианты, о которых я сейчас и расскажу.

Начнём с консольного интерфейса, доступного по команде *Maxima*; он выполнен в традиционном стиле командной строки: на экране чередуются вводимые вами команды и ответы системы на них (рис. 1). Интерфейс, как видите, достаточно незамысловат, но тем не менее все формулы, даже достаточно сложные, вполне читабельны. Графические возможности в чистой консоли недоступны совсем: графики функций, к примеру, могут быть изображены только всё теми же текстовыми символами. Если же запустить консольную Максиму в X-терминале, то графики могут отображаться в отдельных окнах – так же, как и в любом из графических интерфейсов. Единственный реальный плюс консольного интерфейса – это минимальные требования к ресурсам. В остальном всё, как видите, довольно аскетично.

Самый примитивный из графических интерфейсов, – *XMaxima* (рис. 2). На иллюстрации верхняя половина окна – это собственно рабочая область, нижняя – помощь. Кроме этого отдельного окна помощи *XMaxima* практически ничем не отличается от консольного собрата, если тот запущен в X. Посему и тут долго задерживаться не будем.

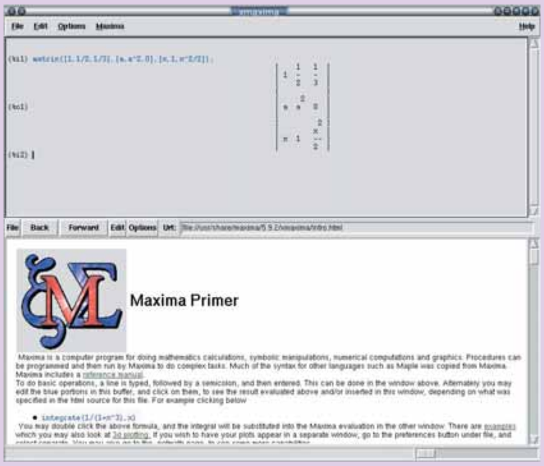


Рисунок 2. *Xmaxima*.

А рассмотрим следующий интерфейс – *Maxima-Emacs*. Он, как нетрудно догадаться, запускает сессию Максимы в буфере широко известного редактора *Emacs*. В результате вызова в редакторе команды *Maxima* (M-x *Maxima*) создаётся новый буфер по имени **Maxima**, в котором и запускается сессия. После этого становятся доступными довольно многочисленные команды взаимодействия с *Maxima* (рис. 3). Привязав эти команды к клавишам на свой вкус, приверженцы этого мега-редактора смогут получить внутри него довольно-таки удобный и богатый возможностями интерфейс (надо сказать, многие команды привязаны к определённым клавишам сразу, но не факт, что умолчательная привязка всем понравится, тем более, что речь о таких любителях настройки всего и вся под свой комфорт, как пользователи *Emacs*). К примеру, на клавишу «Tab», которая в режиме *Maxima* не задействована, можно повесить команду *Maxima*-complete – и получить на привычном месте полноценное автодополнение (по умолчанию эта команда подвешена на M-Tab (Alt-Tab), что многим может быть неудобно, так как эта комбинация, как известно, часто бывает назначена на переключение между окнами). Правда, этот интерфейс также лишён графической отрисовки формул, но все графические возможности самой *Maxima* в нём, в случае запуска в X-версии редактора, естественно, доступны. Кроме того, интересен он не столько сам по себе, сколько во взаимодействии ещё с одним интерфейсом, о котором чуть ниже.

Следующие два интерфейса – *EMaxima* и *iMaxima* – также являются режимами редактора *Emacs*. Первый – скорее не самостоятельный режим, а надстройка над режимом *LaTeX*, которая наверняка понравится тем, кто использует *Emacs* для редактирования *LaTeX*-документов. В отличие от режима *Maxima*, который предназначен для обычного изолированного запуска полноценной *Maxima*-сессии, здесь речь идёт о возможности вставлять отдельные команды *Maxima* и, естественно, результаты их вычислений, прямо в редактируемый *LaTeX*-документ. Для работы этого режима понадобится также расширение tex-site; для *Emacs* оно входит в пакет *auctex*, для *XEmacs21* – в пакет *xEmacs21*-basesupport. Вызывается режим, как обычно, соответственно его названию – командой *EMaxima*-mode (M-x *EMaxima*-mode). Возможности этого режима достаточно богаты, прочитать о них (точно так же как и о режиме *Maxima*) можно в *Maxima-book*, которая входит в состав стандартной документации, находящейся, в зависимости от дистрибутива, в пакете *Maxima* или *Maxima*-doc. Эта часть может быть также доступна в виде отдельного файла; например, в Debian это **/usr/share/doc/Maxima-doc/EMaximaIntro.ps.gz**. Остальную информацию в *Maxima*-book я вам читать не советую – она всё-таки очень устарела (обновлена 19 сентября 2004 года); лучше обратиться к info-страницам или html-документации, которые доступны всё в том же пакете *Maxima* либо *Maxima*-doc, а последняя ещё и на сайте проекта. К примеру, в простейшем случае вы можете создать ячейку *Maxima* комбинацией C-c C-o («o» от фразы «open cell»), ввести в ней любую команду или набор команд Максимы в простой текстовой нотации и получить результат вычисления этой команды либо в обычном текстовом виде нажатием C-c C-u c, либо в *LaTeX*-виде с помощью C-c C-u C (т.е. Ctrl-c Ctrl-u Shift-c). Здесь «u c» происходит от «update cell»; а смежные команды, генерирующие вывод в простой текстовой форме и в форме *LaTeX*, всегда привязаны в Emaxim'е к одинаковым строчной и заглавной буквам соответственно.

Последний *Emacs*-интерфейс к *Maxima* – *iMaxima* – отличается от остальных рассмотренных в этот раз самостоятельным (а не посредством *LaTeX*-документа, как в *EMaxima*) графическим представлением математических формул. Собственно, именно для этого он и создан, и единственная его функциональность заключается именно в отображении в графическом виде TeX-кода, генерируемого *Maxima*. Этот режим можно настроить таким образом, чтобы внутри него запускался режим *Maxima* (т.е. *Maxima-Emacs*), и пользоваться всеми командами послед-

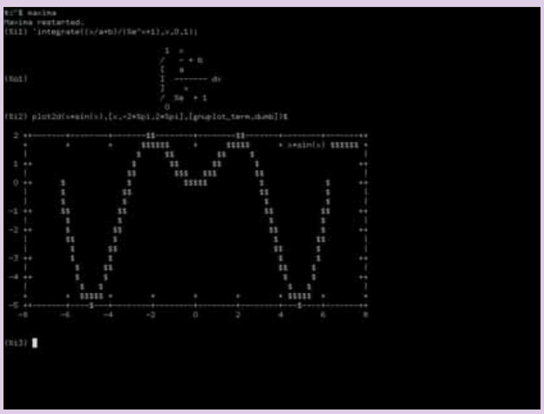


Рисунок 1. Консольная *Maxima*.

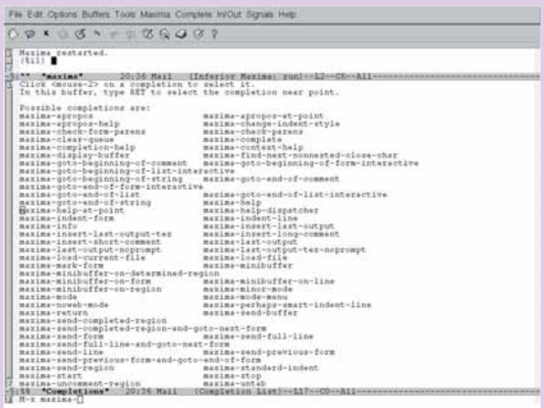


Рисунок 3. *Maxima-Emacs*.

средство – блокировка вычислений. Здесь всё достаточно просто и дополнительно стоит остановиться только на одном моменте. Если апострофом предварён вызов функции (встроенной ли, пользовательской – несущественно), то блокируется вычисление самой функции, но не её аргументов. Если же поставить апостроф перед выражением, заключённым в скобки, то невычисленными останется всё это выражение целиком, т.е. и все входящие в него функции, и все аргументы этих функций. Например:

(%i1) $a: \frac{\pi}{4} \$ \, b: \frac{\pi}{2} \$$

(%i3) $\mathrm{integrate}(\sin(x), x, a, b)$

(%o3) $\frac{\sqrt{2}}{2}$

(%i4) $'\mathrm{integrate}(\sin(x), x, a, b)$

(%o4) $\int_{\frac{\pi}{4}}^{\frac{\pi}{2}} \sin x \, \mathrm{d}\, x$

(%i5) $'(\mathrm{integrate}(\sin(x), x, a, b))$

(%o5) $\mathrm{integrate}(\sin x, x, a, b)$

В противовес блокировке вычислений, можно также принудительно вычислить любое выражение – для этого тоже существует оператор, состоящий из двух апострофов:

(%i6) $''\%$

(%o6) $\frac{\sqrt{2}}{2}$

В терминологии *Maxima* невычисленная форма выражения называется «noun form», вычисленная – «verb form». Сохраняя лингвистические параллели, на русский я бы это перевёл как «несовершённая форма» и «совершённая форма».

Если говорить о ячейках ввода-вывода, то значение ячейки ввода в *Maxima* закономерно сохраняется до его вычисления (т.е. в несовершённой форме), а значение ячейки вывода – после (т.е. в совершённой); другими словами, тут сохраняется естественный порядок «ввод –> вычисление –> вывод».

(%i7) $\%i5$

(%o7) $'(\mathrm{integrate}(\sin(x), x, a, b))$

(%i8) $''\%i5$

(%o8) $\mathrm{integrate}(\sin(x), x, a, b)$

(%i9) $''(''\%i5)$

(%o9) $\frac{\sqrt{2}}{2}$

Как видите, операторы вычисления и блокировки вычислений имеют накопительный эффект. О другой стороне этого эффекта мы поговорим чуть ниже.

Оператор, обозначенный двумя апострофами, является синонимом к функции **ev**(выражение). Сама функция **ev** предоставляет гораздо более широкие возможности, нежели простое принудительное вычисление заданного выражения: она может принимать произвольное число аргументов, первый из которых – вычисляемое выражение, а остальные – специальные опции, которые как раз и влияют на то, как именно будет производиться вычисление. Точно так же, как двойной апостроф – сокращение для ev без дополнительных опций, есть ещё более упрощённая запись функции ev с опциями: в этом случае вместо имени функции и скобок вообще ничего писать не нужно; т.е. «**ev(выражение, опц1, опц2, ...)**» можно записать просто как «**выражение, опц1, опц2, ...**».

Первая из таких опций связана с автоупрощением. Глобально автоупрощение регулируется переключателем **simp** (от «simplification» – упрощение), и по умолчанию оно включено; в любой момент его можно выключить, установив значение переключателя в *false*. Опция функции ev, одноимённая этому переключателю, позволяет включить упрощение для данного конкретного вычисления – вне зависимости от того, включено или выключено оно глобально:

>>

ИНТЕРФЕЙСЫ К MAXIMA (ПРОДОЛЖЕНИЕ)

него и их клавиатурными привязками. Т.е. фактически режим *iMaxima* в таком варианте можно рассматривать как графический интерфейс уже над *Maxima-Emacs*; именно это может добавить дополнительной привлекательности последнему. В отличие от всех рассмотренных выше интерфейсов, *iMaxima* – сторонний проект, разрабатываемый отдельно; начиная со второй половины прошлого года iMaxim'ой занимается новый автор, и на данный момент проект активно развивается. Для его установки вам понадобится дополнительно установить пакет breqn, отвечающий за перенос строк в математических формулах в формате *LaTeX*. Инструкцию по установке самой *iMaxima* и *breqn* вы можете найти на сайте проекта (адрес см. во врезке).

Кроме всех рассмотренных, существуют ещё два интерфейса к Максиме: *Symaxx* и *Kayali*. Но учитывая, что оба проекта довольно давно не подают признаков жизни (*Kayali* находится на стадии альфа-версии, последнее обновление которой вышло 6 июня 2005 года; а *Symaxx* не обновлялся с 17 декабря 2001 года), то они не достойны большего, чем просто упоминание. Кроме того, существуют полноценные web- и telnet-интерфейсы к *Maxima*, благодаря которым вы можете поработать с нею, не имея даже её у себя на компьютере, прямо через Интернет.

Как видите, способность *Maxima* взаимодействовать с внешними интерфейсами используется достаточно широко – есть из чего выбрать тот интерфейс, который лучше всего подойдёт именно вам.

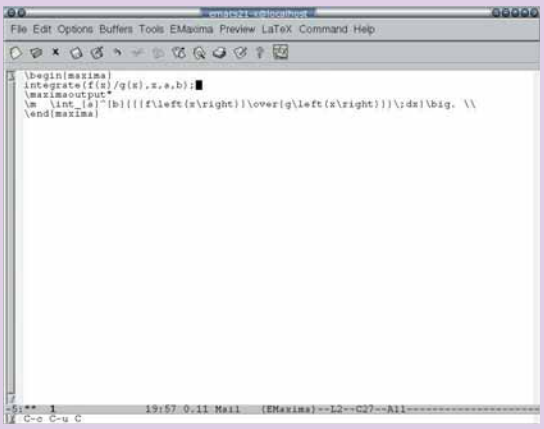


Рисунок 4. Ячейка Максимы введенная и вычисленная в Emacs-режиме *Emaxima*.

АДРЕСА В ИНТЕРНЕТЕ

Сайт проекта Maxima: **http://maxima.sourceforge.net/**
Сайт iMaxima: **http://members3.jcom.home.ne.jp/imaxima/**
Сайт Kayali: **http://kayali.sf.net/**
Сайт Symaxx: **http://symaxx.sf.net/**
Web-интерфейс к Максиме: **http://www.et.byu.edu/~koj/maxima.html**

(%i1) $\mathrm{simp: false\$}$

(%i2) $\left(\frac{1}{2}+\frac{2}{3}\right)\left(\frac{3}{2}+\frac{4}{5}\right);\ \frac{122!}{112!};\ \sqrt{x^2}$

(%o2) $\left(\frac{1}{2}+\frac{2}{3}\right)\left(\frac{3}{2}+\frac{4}{5}\right)$

(%o3) $\frac{122!}{112!}$

(%o4) $\sqrt{x^2}$

(%i5) $\%o2, \mathrm{simp};\ \%o3, \mathrm{simp};\ \%o4, \mathrm{simp}$

(%o5) $\frac{161}{60}$

(%o6) $5001271160299622208 00$

(%o7) $|x|$

Тут нужно отметить ещё, что вызов **kill(all)** не восстанавливает умолчательные значения переключателей; т.е. если мы, к примеру, изменили значение переключателя simp, как в примере выше, то для того, чтобы вернуться к изначальному порядку вещей, установленному сразу после запуска *Maxima*, нам нужно не только сделать **kill(all)**, но и вручную назначить **simp:true**.

Опция *diff* принудительно раскрывает все производные и полные дифференциалы; а опция **derivlist(x, y, ..., v)** – производные относительно переменных, заданных в качестве её аргументов, а также полные дифференциалы (так как они не зависят ни от каких переменных):

(%i1) $f\mathrm{: 'diff}(w^n, w) + \mathrm{'diff}\left(\frac{y}{x}, x, 1, y, 1\right)\$$

(%i2) $\mathrm{ev}(f, \mathrm{derivlist}(x, y))$

(%o2) $\frac{d}{d\,w}w^n - \frac{1}{x^2}$

(%i3) $\mathrm{ev}(f, \mathrm{derivlist}(y, v, w))$

(%o3) $\frac{d}{d\,x}\left(\frac{1}{x}\right) + n\,w^{n-1}$

(%i4) $\mathrm{ev}(f, \mathrm{diff})$

(%o4) $n\,w^{n-1} - \frac{1}{x^2}$

Как видите, если из нескольких переменных из **diff** в **derivlist()** заданы не все, то раскрывается производная только по заданным переменным; это и понятно, так как выражения **diff(f, x, 1, y, 1)**, **diff(diff(f, x), y)** и **diff(diff(f, y), x)** математически эквивалентны [по крайней мере, для «хороших» функций, – прим.ред]. Если же аргумент опции **derivlist()** вообще не является переменной дифференцирования, он просто игнорируется.

Опция nouns раскрывает вообще все несовершённые формы – и производные в том числе:

(%i5) $\mathrm{ev}(f, \mathrm{nouns})$

(%o5) $n\,w^{n-1} - \frac{1}{x^2}$

Опция *float* преобразовывает все рациональные числа в конечную десятичную запись; опция numer включает опцию *float* и, кроме того, приводит к десятичному виду многие математические функции от числовых аргументов:

(%i1) $\cos(1)$

(%o1) $\cos 1$

(%i2) $\%, \mathrm{numer}$

(%o2) 0.54030230586814

Опция **noeval** блокирует сам этап вычисления как таковой; т.е. её можно использовать для того, чтобы применить к выражению другие опции функции ev, не перевычисляя его. При этом опять-таки нужно понимать разницу между вычислением и упрощением:

(%i1) $\cos\left(\frac{\pi}{4}\right), \mathrm{noeval}$

(%o1) $\frac{\sqrt{2}}{2}$

(%i2) $\mathrm{simp: false\$\ ev}\left(\cos\left(\frac{\pi}{4}\right)\right)$

(%o3) $\cos\left(\frac{\pi}{4}\right)$

Таким образом, мы можем принудительно упростить выражение, не перевычисляя его.

Опция **eval** – напротив, проводит дополнительно ещё один процесс вычисления. Здесь стоит поговорить подробнее о накопительном эффекте вычисления, который я уже демонстрировал выше. Так как в Максиме значениями символов могут выступать самые разнообразные выражения, то в эти выражения тоже могут входить некоторые символы, которые тоже могут иметь свои значения; и такая цепочка «вложенных значений» может продолжаться сколь угодно глубоко. Один вызов функции **ev** (без опции **eval**) опускается по этой цепочке в глубину на один уровень:

(%i1) $y\mathrm{:}\ x^n\$\ n\mathrm{:}\ k+m\$\ k\mathrm{:}\ 2\ p\$\ p\mathrm{:}\ 3\$$

(%i5) y

(%o5) x^n

(%i6) $\mathrm{ev}(y)$

(%o6) x^{m+k}

(%i7) $\mathrm{ev}(\mathrm{ev}(y))$

(%o7) $x^{2\,p+m}$

(%i8) $\mathrm{ev}(y), \mathrm{eval}$

(%o8) x^{m+6}

Напомню, что здесь **ev(y), eval** является сокращённой записью от **ev(ev(y), eval)**, таким образом вычисление в этом выражении проводится трижды. Кроме того, хочу обратить ваше внимание на порядок назначения выражений символам; здесь существенно, что на момент задания каждого выражения входящий в него символ ещё не был определён – иначе в выражение автоматически подставлялся не сам символ,

а его значение. Таким образом, если бы мы произвели эти же назначения в обратном порядке, то значением символа y стало бы x^{m+6} – безо всяких принудительных вычислений.

В продолжение разговора о накопительном эффекте и «цепочных» вычислениях придётся кстати переключатель *infeval*. Он заставляет ev перевычислять выражение до тех пор, пока оно не перестанет изменяться при последующих вычислениях. В частности, этот переключатель можно использовать и для того, чтобы разблокировать блокировку вычислений любой глубины вложения:

(%i1) $'('('('(\mathrm{integrate}(x^2,x,0,1)))))$

(%o1) $'\Big('\big('\big(\mathrm{integrate}(x^2,x,0,1)\big)\big)\Big)$

(%i2) $\mathrm{ev}(\%, \mathrm{infeval})$

(%o2) $\frac{1}{3}$

В других ситуациях использовать этот переключатель следует с осторожностью: не забывайте, что он может привести к зацикливанию.

О других константных опциях и переключателях функции ev можно узнать из **? ev** и **? evflag**, а мы наверняка ещё рассмотрим многие из них позже, когда они будут более актуальны в контексте повествования.

Кроме константных значений есть ещё несколько видов опций. Первый из них – это имена специальных функций, которые занимаются упрощением или преобразованием математических выражений. Будучи упомянута по имени в качестве опции, такая функция просто применяется к вычисляемому выражению. Например, **выражение, fullratsimp** – это то же самое, что и **fullratsimp(ev(выражение))**. Полный список таких функций вы можете найти в **? evfun**.

Если в качестве опции ввести имя любой другой функции, не имеющей свойства **evfun**, то все несовершённые вхождения этой функции будут заменены совершёнными, т.е. принудительно вычислены.

Также в качестве опции можно задать назначение символа или функции; все такие назначения действуют локально в пределах вычисляемого выражения, и все подстановки производятся параллельно:

(%i1) $x+y+z, x{:}\,1, z{:}\,a, y{:}\,x\,z$

(%o1) $x\,z+a+1$

Опция подстановки символа допустима не только в виде оператора присвоения, но и в виде равенства; сделано это, в частности, для того, чтобы в качестве подстановок можно было использовать решения, найденные функцией **solve**:

(%i1) $\mathrm{ev}(y^x, \mathrm{solve}(x^3+3\,x^2+3\,x+1))$

(%o1) $\frac{1}{y}$

Вот и всё на сегодня. В следующий раз мы начнём с уже упомянутых вскользь функций по упрощению и преобразованию выражений. **LXF**

ЧЕРЕЗ МЕСЯЦ

Мы упростим себе жизнь при помощи соответствующих функций Maxima

Третье ИЗМЕРЕНИЕ

(окончание. Начало на стр. 68–70)

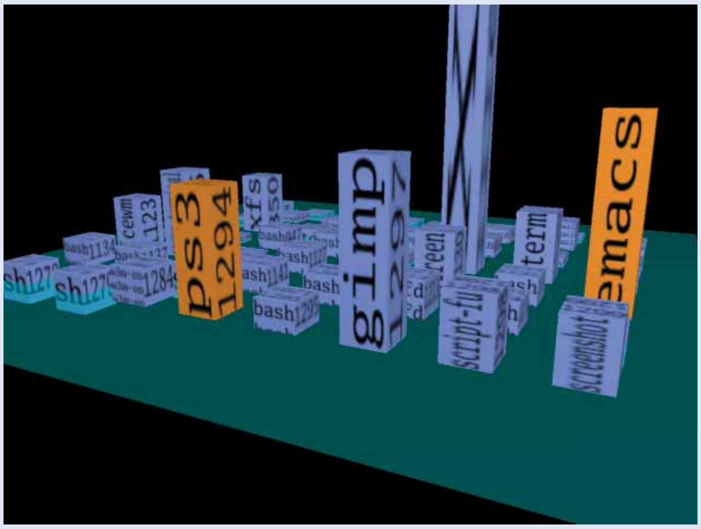


Рисунок 4. *ps3* выводит таблицу процессов в очень наглядной форме.

который также может быть использован как настольная среда. Построенный по клиент-серверной схеме (подобно X-серверу), сервер *s3d* запускается на компьютере пользователя и ожидает запросов от клиентов на вывод содержимого приложений, которые должны быть преобразованы на сервере.

Вывод информации о процессах

Linux является многозадачной системой, в которой одновременно выполняются сотни процессов. Для их вывода информации о запущенных процессах традиционно используются утилиты ps и top или какой-либо графический инструмент. Утилита *ps3* (см. ссылку 7) выводит таблицу процессов в виде блоков на вращающемся диске (*Рис. 4*), подобно тому, как *TDFSB* показывает файловую систему. Перемещаясь при помощи клавиатуры, можно рассмотреть имя процесса, если подойти к выбранному столбцу «с другой стороны», то будет показан номер PID. Высота сообщает, сколько памяти занимает процесс, а цвет – сколько он потребляет ресурсов процессора. Поэтому, запустив утилиту, сразу можно увидеть какой из процессов самый «жадный». Относительные пропорции выводимого изображения и направления вращения можно изменять при помощи мыши. Для этого, необходимо, придержав кнопку мыши двигать ее вниз/верх (пропорции) или вправо/влево (вращение). Кроме того, можно использовать предустановленный режим вывода информации, доступный по клавишам F1-F6. Практически аналогичные возможности предоставляет и другая утилита *GLload* (см. ссылку 8). В командной строке или по щелчку правой кнопкой в контекстном меню можно дополнительно вывести дату, время, метку и внешний вид.

Это далеко не все проекты. Так, например *glbiff* выведет крутящийся рисунок при поступлении почтового сообщения полученного посредством *fetchmail*.

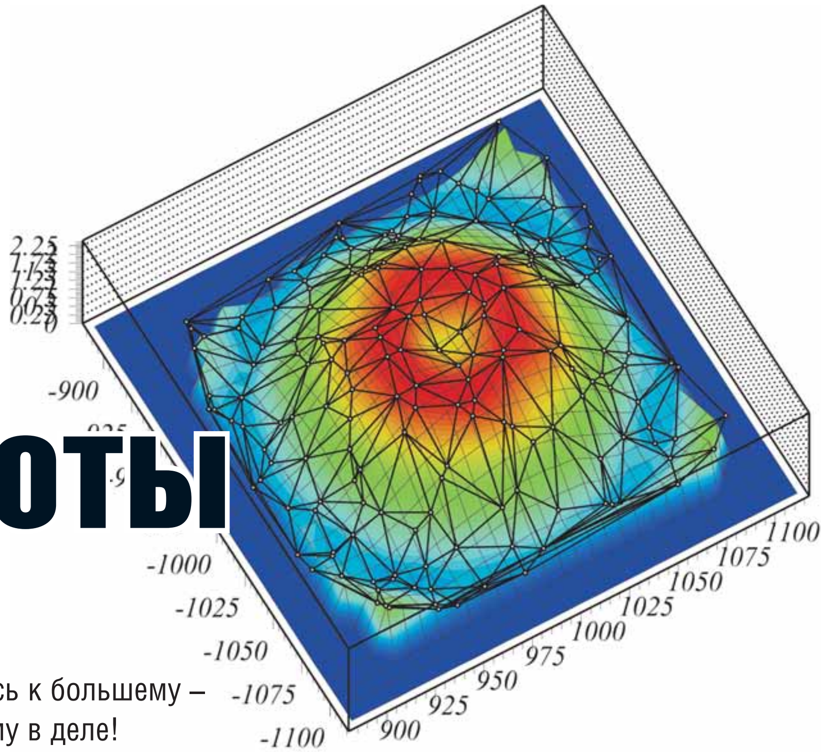
Инструментов, выводящих информацию в трехмерном виде, предостаточно и поэтому в Linux уже сегодня можно представить, как возможно будут выглядеть рабочие столы пользователей будущего. **LXF**

ПОЛЕЗНЫЕ ССЫЛКИ

1. Сайт проекта 3D-Desktop – http://desk3d.sourceforge.net/
2. Сайт проекта Metisse -http://insitu.lri.fr/~chapuis/metisse/
3. Сайт проекта fsv – http://fsv.sourceforge.net
4. Сайт проекта TDFSB – http://www.determinate.net/webdata/seg/tdfsb.html
5. Библиотека Nucleo – http://insitu.lri.fr/~chapuis/software/metisse/nucleo-0.1-20041216.tar.bz2
6. Сайт Window Manager Icons – http://wm-icons.sourceforge.net/
7. Сайт проекта ps3 – http://user.tninet.se/~uhu537u/ps3.html
8. Утилита Glload – http://reality.exsgi.org/eile/misc/glload-0.4.2.tar.gz
9. Сайт FileCityMap – http://thisiselliot.homeip.net/
10. Сайт s3d – http://s3d.berlios.de/
11. Сайт True3D*Shell – http://www.sixtyfourbit.org/3dshell.htm

АНАЛИЗ ДАННЫХ

PAW: приемы работы



ЧАСТЬ 2 Впечатлены возможностями PAW? Приготовьтесь к большему – сейчас **Евгений Балдин** продемонстрирует вам эту систему в деле!

МЕСЯЦ НАЗАД

Мы начали знакомство с PAW с простейших команд и обзора возможностей этого пакета

Система анализа данных *PAW* или Physics Analysis Workstation для своей работы не требует досконального знаний всех подсистем и команд. Но чтобы действовать эффективно, следует изучить логику построения команд и стандартные приемы. Это позволит в дальнейшем легко получать навыки для выполнения более сложных задач.

Если даже вы не планируете использовать *PAW*, в любом случае будет полезно присмотреться к этому инструменту, так как он сравнительно прост и основные концепции, необходимые для анализа данных, достаточно прозрачны для понимания. Создатели *PAW* действовали по принципу минимализма. Делалось только необходимое – никаких «рюшечек», зато просто. Жесткая структура команд дополнена возможностью писать свои функции и скрипты.

Простейший анализ

Учиться лучше всего на примере реального анализа. Попробуем сделать нечто подобное.

Будем считать, что программа *PAW* уже запущена и мы находимся в рабочей директории. Вызов внешних команд обеспечивается с помощью инструкции SHELL (можно сократить до sh).

```
PAW > sh ls
ascii.png  lkravg.dat   PAW.metafile   ee-ang.rz
th1.eps    last.kumac   last.kumacold  sin.dat
```

Необходимо провести предварительный анализ данных, представленных в текстовом файле **lkravg.dat**:

```
1099279655  4119 0.8318 0.0014 1.13 5.99195
1099397693  4126 0.8404 0.0032 1.07 6.001685
  …
```

Колонки соответствуют **time_t** – времени в секундах, номеру измерения, исследуемому значению, ошибке значения для текущего измерения и двум сторонним параметрам, от которых интересующее нас значение может зависеть. Задача: имея время и два сторонних параметра, попробовать предсказать исследуемое значение.

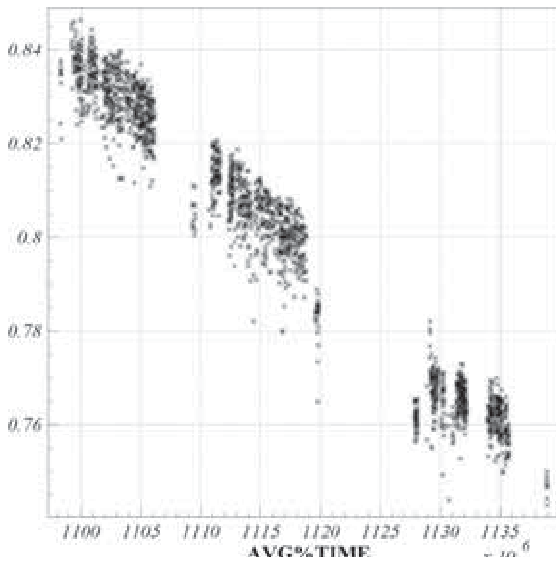
Анализировать можно и без модели явления, но чтобы правильно подготовить данные для исследования, необходимо ее иметь. Легенда для этих данных следующая: исследуемое значение представляет из себя степень «ухудшения» качества жидкого криптона (LKr – Liquid Kripton) в LKr-калориметре для регистрации энергии элементарных частиц. Качество вычисляется в относительных единицах по амплитуде сигнала от космических мюонов (эти частицы хорошо выделяются и есть всегда). Два параметра, от которых может зависеть качество: избыточное давление LKr и магнитное поле, в котором находится калориметр. Избыточное давление примерно линейно связано с температурой LKr, что влияет на амплитуду сигнала. В магнитном поле прямолинейная траектория мюона искажается, что тоже может влиять на амплитуду.

В первую очередь, необходимо прочитать данные из текстового файла. Для этого можно воспользоваться командой **VECTOR/READ** (help ve/re):

```
#чтение текстового файла в вектора
PAW > ve/re time,run,avg,avg_er,P,H lkravg.dat
# меняем тип маркера
PAW > set mtyp 2
# изобразим зависимость исследуемого значения от времени
PAW > ve/pl avg%time
```

Из рисунка ниже видно, что исследуемое значение в среднем уменьшается за большой промежуток времени (исследуемый интервал равен году и четырем месяцам), и в то же время у данных есть структура, соответствующая более короткому интервалу.

Сначала исключим временную зависимость. Воспользуемся для этого стандартной процедурой подгонки для векторов **VECTOR/FIT X Y EY FUNC** (help ve/fit), где **X** соответствует оси времени, **Y** – исследуемо-



Зависимость качества LKr от времени.

му значению, **EY** – ошибки определения значения, а **FUNC** – подгоночной функции. Вместо **FUNC** можно передать любую функцию, написанную на FORTRAN, но в нашем случае достаточно полинома первой степени. Для полинома в *PAW* есть сокращенное обозначение **PN**, где **N** – степень полинома.

```
#подгоняем временную зависимость прямой
PAW > ve/fit time avg avg_er P1
 **********************************************************
 *                                                        *
 *      Function minimization by SUBROUTINE HFITV         *
 *      Variable-metric method                            *
 *      ID =        0  CHOPT =                            *
 *                                                        *
 **********************************************************
Convergence when estimated distance to minimum (EDM) .LT. 0.10E+01
EXT PARAMETER             APPROXIMATE      STEP          FIRST
NO.  NAME   VALUE          ERROR           SIZE          DERIVATIVE
1     P1    3.3620         0.50070E-03    0.85051E-02   -223.19
2     P2    -0.22939E-08   0.44872E-12    0.58032E-11   -0.27325E+12
 CHISQUARE = 0.1680E+02  NPFIT =  1644
```

При выполнении процедуры подгонки *PAW* выдает стандартную «портянку». Основной интерес для пользователя представляют результаты подгонки. В случае полинома первой степени подгоняются два параметра **P1** – свободная константа и **P2** – коэффициент пропорциональности

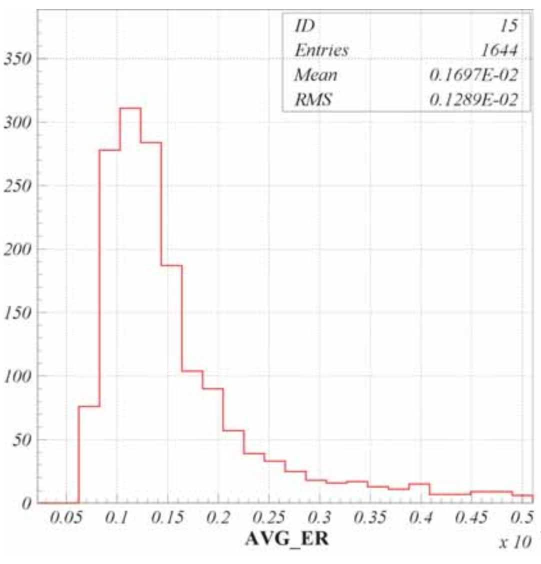
Для исключения временной зависимости воспользуемся пакетом *SIGMA* (**help sigma**). С помощью команды **sigma** векторами можно манипулировать, как обычными переменными:

```
#создаем новый вектор уже без временной зависимости
PAW > sigma avg1=avg-3.3620+0.22939E-08*time
```

Прежде чем действовать дальше, оценим, какую точность в принципе можно ожидать. Вектору исследуемой величины **avg** соответствует вектор ошибок **avg_er**. Посмотрим, чему равны эти ошибки. Число измерений превосходит полторы тысячи, поэтому просмотреть все значения глазами не очень реально. Лучше создать гистограмму:

```
#создаем из значений вектора гистограмму N15
PAW > ve/pl avg_er 15
#включаем отображение статистики (число событий/среднее/разброс)
PAW > opt sta
#меняем цвет гистограммы на красный
PAW > set hcol 2
#вывод гистограммы N15 в диапазоне от 0. до 0.01
PAW > hi/plot 15(0.:0.01)
```

Из гистограммы видно, что большинство измерений имеют ошибку меньше $0.15 \cdot 10^{-2}$–$0.2 \cdot 10^{-2}$. То есть, даже при самом удачном раскладе точность предсказания не будет выше этих 0.15%-0.2%.



Гистограмма ошибок измеряемого параметра.

После того, как мы убрали основную (временную) зависимость, можно проверить, зависит ли изучаемая переменная от давления (P) и магнитного поля (H). Для этого воспользуемся способностью *PAW* создавать и отображать двумерные гистограмм:

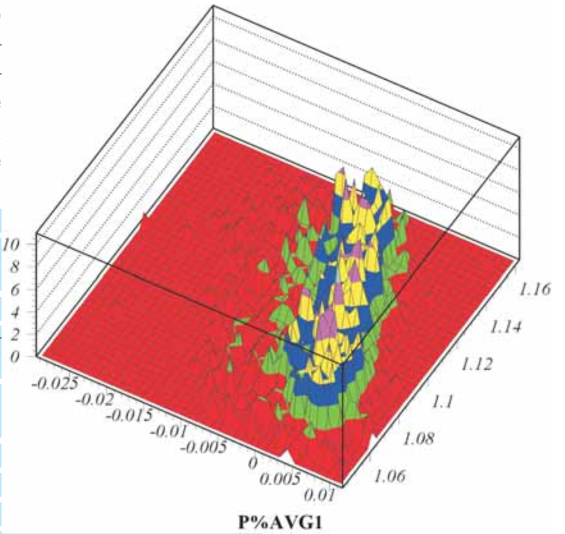
```
#создаем двумерную гистограмму: давление от avg1
PAW > ve/pl P%avg1 10
#изобразим двумерную гистограмму 10 в виде поверхности
PAW > SURF 10 65 -30 1
```

Обратите внимание на разделитель **%** между векторами в команде **VECTOR/PLOT**. Значениям первого вектора противопоставляется ось ординат (ось Y), а значениям второго ось абсцисс (ось X).

Команда **HISTOGRAMgfv/2D_PLOT/SURFACE** [**ID THETA PHI CHOPT**] (**help surf)** позволяет отобразить двумерную гистограмму в виде поверхности. Здесь **ID** – номер гистограммы, **THETA** и **PHI** – углы поворота гистограммы θ и φ в сферической системе координат, **CHOPT** – опции изображения.

Из картинки справа видно, что какая-то зависимость есть – избавимся от нее, как это было сделано с временной зависимостью. В результате подгонки сначала для давления, а потом для поля можно получить еще две формулы:

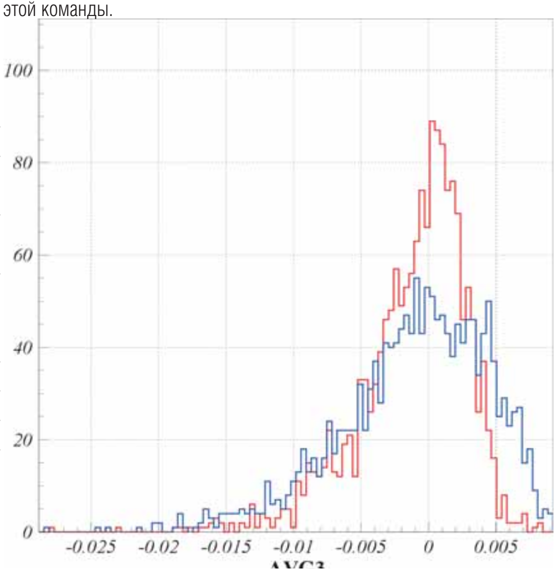
```
#поправка на давление
PAW > sigma avg2=avg1-0.10901+0.99336E-01*P
#поправка на магнитное поле
PAW > sigma avg3=avg2+0.10844E-01-0.18258E-02*H
#сравниваем две гистограммы до и после поправок
#на давление и магнитное поле
#переключаем цвет на красный
PAW > set hcol 2
#рисуем гистограмму по значениям вектора
PAW > ve/plot avg3
#переключаем цвет на синий
PAW > set hcol 4
#рисуем вторую гистограмму поверх ‘s’ - superimpose
PAW > ve/plot avg1 ! ‘s’
```



Гистограмма ошибок измеряемого параметра.

Обратите внимание на команду **SET** (**help GRAPHICS/SET**). Эта инструкция позволяет менять параметры графического представления данных. Если запустить ее без опций, то будет выдан список переменных, которые устанавливаются с помощью этой команды.

На рисунке справа представлены две гистограммы. Красная – конечный результат, а синяя – то, что осталось после исключения временной зависимости. Очевидно, что после учета давления и поля разброс уменьшился. Несимметричность гистограмм указывает на то, что временная зависимость на самом деле нелинейная. В реальности при подгонке использовалась экспоненциальная зависимость плюс некоторая константа. Выбор подгоночной функции был продиктован физической моделью явления, но по большому счету на этом временном интервале она не сильно отличается от обычной прямой.



Сравнение двух гистограмм с помощью наложения.

В результате всех действий была получена зависимость:

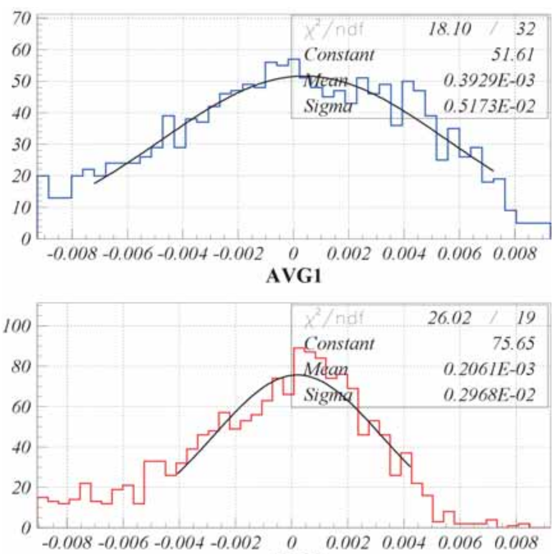
AVG=3.46+2.29•10-9^{-9}∗time-9.9∗10-2^{-2}P+1.8∗10-3^{-3}H

Эта функция позволяет предсказывать исследуемое

значение в зависимости от времени, давления и магнитного поля, но какова точность этого предсказания? Для ее оценки достаточно получить распределение разницы значений между экспериментальными точками и данной зависимостью, а затем взять его ширину. Можно просто оценить ширину распределения «на глазок», посмотрев на обсуждаемый рисунок, а можно получить ее из известной функции, более-менее описывающей это распределение, например, из функции Гаусса:

```
#создаем из значений вектора avg1 гистограмму N11
PAW > ve/plot avg1 11
#подгоняем гистограмму N11 в диапазоне (-0.007,0.007)
#функцией Гаусса (G - Gauss)
PAW > hi/fit 11(-0.007:0.007) G
        ...
EXT    PARAMETER                                    ...
NO.    NAME      VALUE            ERROR             ...
 1     Constant  51.608           1.9494            ...
 2     Mean      0.39292E-03      0.20076E-03       ...
 3     Sigma     0.51728E-02      0.27301E-03       ...
        ...
#создаем из значений вектора avg3 гистограмму N13
PAW > ve/plot avg3 13
PAW > hi/fit 13(-0.004:0.004) G
        ...
EXT    PARAMETER                                    ...
NO.    NAME      VALUE            ERROR             ...
 1     Constant  75.655           3.0477            ...
 2     Mean      0.20614E-03      0.11857E-03       ...
 3     Sigma     0.29684E-02      0.16655E-03       ...
        ...
#делим графическое окно на две зоны по оси ординат (help zone)
PAW > zone 1 2
#меняем цвет гистограммы на синий
PAW > set hcol 4
#рисуем гистограмму в диапазоне (-0.009,0.009)
PAW > hi/pl 11(-0.009:0.009)
#меняем цвет гистограммы на красный
PAW > set hcol 2
PAW > hi/pl 13(-0.009:0.009)
```

Из всех значений в данном случае интересно только значение ширины распределения, или SIGMA[1].



Сравнение двух гистограмм. Подгонка функцией Гаусса.

Если учитывать только временную зависимость, то точность предсказания будет примерно 0.52%, если же учесть давление и магнитное поле, то точность улучшится до 0.30%, что существенно хуже идеальных 0.15-0.2%. Очевидно, что остались еще какие-то неучтенные систематики.

Оценим, какую систематику удалось выбрать, учтя зависимость от давления и магнитного поля. Для этого воспользуемся системой *COMIS* (help comis):

```
PAW > comis
 PAW
CS> type sqrt(0.52**2-0.30**2)
MND> end
 *T  SQRT(0.52**2-0.30**2) = 0.4247352
 PAW
CS> end
```

В общем, неплохо. Кстати, более тщательный анализ никаких кардинальных улучшений не дал – удалось только уменьшить «хвосты» и сделать итоговое распределение более симметричным за счет выбора более сложной подгоночной функции.

Быстрый анализ позволяет оценить, к какой точности имеет смысл стремиться. Это очень важно, так как сложность получения большей точности увеличивается от требуемой точности существенно нелинейным образом. Фактически на только что изложенный анализ по разным причинам ушел примерно месяц реального времени. Правда, основные проблемы были вовсе не технические. В частности очень много времени ушло на осознание, что исследуемая величина зависит от времени – это не казалось очевидным.

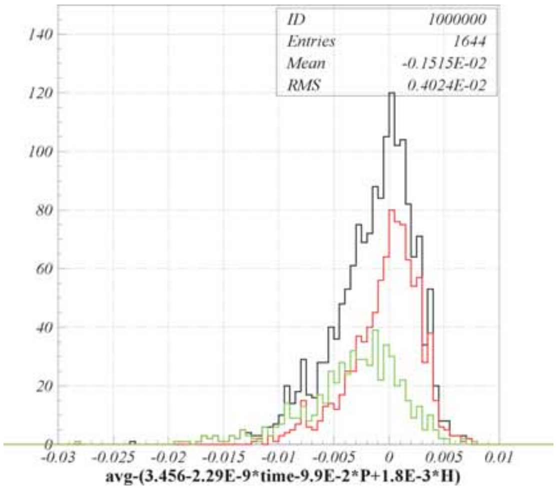
Ntuple

То, что только что было сделано с помощью векторов можно проделать с помощью ntuple. Для этого надо сначала создать *ntuple* (**NTUPLE/CREATE**), а затем считать в него текстовый файл (**NTUPLE/READ**).

```
# создаем ntuple с ID=1
PAW > nt/cre 1 'LKr quality' 6 ! ! time run avg er P H
#читаем в ntuple с ID 1 текстовый файл
PAW > nt/read 1 lkravg.dat
 ==>  1644 events have been read
PAW > nt/print 1
******************************************************************
* NTUPLE ID=     1     ENTRIES=     1644      LKr quality
******************************************************************
*  Var numb  *  Name   *  Lower            *  Upper            *
******************************************************************
*     1      * TIME    * 0.109828E+10      * 0.113892E+10      *
*     2      * RUN     * 0.406400E+04      * 0.709400E+04      *
*     3      * AVG     * 0.742800E+00      * 0.846400E+00      *
*     4      * ER      * 0.700000E-03      * 0.201000E-01      *
*     5      * P       * 0.104400E+01      * 0.116000E+01      *
*     6      * H       * 0.000000E+00      * 0.704971E+01      *
******************************************************************
#включаем отображение статистики
PAW > opt stat
#отрисовываем разницу между экспериментом и предсказанием
#с различным ограничением на величину ошибки er
PAW > set hcol 1
PAW > nt/pl 1.avg-(3.456-2.29E-9*time-9.9E-2*P+1.8E-3*H)
PAW > set hcol 2
PAW > nt/pl 1.avg-(3.456-2.29E-9*time-9.9E-2*P+1.8E-3*H) er<0.0015 ! !
! 's'
PAW > set hcol 3
PAW > nt/pl 1.avg-(3.456-2.29E-9*time-9.9E-2*P+1.8E-3*H) er>0.0015 ! !
! 's'
```

Преимущество при использовании *ntuple* заключается в том, что интерактивно можно накладывать условия для фильтрации данных. Как правило, *ntuple* создаются с помощью внешних программ, а *PAW* используется уже для интерактивного анализа. В стандартной документации к *PAW* очень подробно описано, как это делается.

1 В случае гауссоподобного распределения в диапазоне $(\langle x\rangle-\sigma,\langle x\rangle+\sigma)$ лежит примерно 68% от всех событий. $\langle x\rangle$ — среднее значение или MEAN, σ соответствует SIGMA

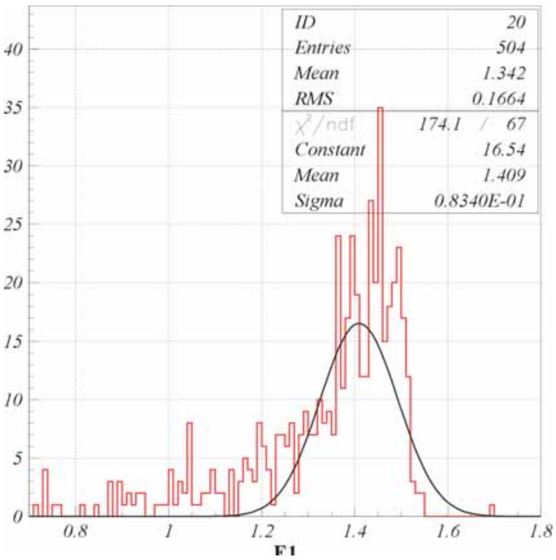


Разность между экспериментом и предсказания в зависимости от наложенного условия на величину ошибки.

Гистограммы

Гистограммы – это базовые объекты *PAW*. Непосредственно перед отображением данные почти всегда преобразуются в одно- или двумерную гистограмму. На гистограмму можно просто смотреть, а можно подгонять какой-либо теоретической зависимостью (**HISTOGRAM/FIT**).

```
#поиск rz-файлов в базовой директории (файл создан внешней программой)
PAW > sh ls *.rz
ee-ang.rz
#чтение файла
#известно что в файле есть ntuple с ID=1
PAW > hi/fil 1 ee-ang.rz
#создаем гистограмм N20 из переменной E1 с некоторыми условиями
PAW > nt/plot 1.E1 f1=11&&f2=-11&&E1<2 ! ! ! ! 20
#подгоняем гистограмму распределением Гаусса
PAW > hi/fit 20 G
```



Подгонка гистограммы с помощью функции Гаусса и пользовательской функции.

Из рисунка видно, что наблюдаемое распределение функцией Гаусса не подгоняется. Надо что-то делать. Очевидно, что теоретическая функция должна быть как минимум несимметрична. Для этой цели подойдет так называемый логарифмический гаусс. Для подгонки надо создать файл **loggaus.for** примерно следующего содержания:

```
C Файл loggaus.for
     real function loggaus(x)
C   С помощью этого common-блока PAW получает доступ
C   к параметрам функции
     common/PAWPAR/PAR(4)
     sqrtln4=1.177410022515475
     A=PAR(1)
     pike=PAR(2)
     sigma=PAR(3)*pike/100.
     assim=PAR(4)
     loggaus=0.
     if (abs(assim).le.1.E-6) then
        assim=sign(1.E-6,assim)
     endif
     if (sigma.le.0.) goto 10
     xx=1.+sinh(assim*sqrtln4)/sqrtln4*(x-pike)/sigma
     if (xx<1.E-07) goto 10
     loggaus=A*exp(-((log(xx)/assim)**2+assim**2)/2.)
10   continue
     end
```

Функция зависит от четырех параметров: **A** – амплитуда, **pike** – местоположение пика, **sigma** – ширины распределения в процентах, **assim** – асимметрии.

```
#создаем вектор параметров с начальными значениями
PAW > ve/cre par(4) r 25. 1.4 5. 0.
# создаем вектор с минимально допустимыми значениями (на глазок)
PAW > ve/cre pmin(4) r 10. 1.3 1. -1.
# создаем вектор с максимально допустимыми значениями
PAW > ve/cre pmax(4) r 40. 1.5 10. 1.
# создаем вектор, для ошибок подгонки
PAW > ve/cre err(4) r
# просим PAW подогнать распределение теоретической функцией
# опции подгонки:
#     B - учитывать минимально/максимально допустимые значения
#     M - перейти интерактивную сессию Minuit
# M обычно не используется, так как действия PAW при подгонке
# по умолчанию вполне разумны
PAW > hi/fit 20 loggaus.for "BM" 4 par ! pmin pmax err
       …
# задаем имена параметров
Minuit > name 1 A
Minuit > name 2 pike
Minuit > name 3 sigma
Minuit > name 4 assim
# задаем метод минимизации (migrad, обычно, самый подходящий)
Minuit > migrad
# просим попробовать улучшить подгонку (дольше, но чуть точнее)
Minuit > improve
       …
Minuit > exit
       …
#смотрим результаты подгонки
PAW > ve/print par
 PAR(1) = 35.4279
 PAR(2) = 1.4809
 PAR(3) = 4.30618
 PAR(4) = -0.999999
#смотрим ошибки
PAW > ve/print err
 ERR(1) = 3.61113
 ERR(2) = 0.00484378
 ERR(3) = 0.339507
 ERR(4) = 0.174973
```

```
#нарисовать гистограмму 20 еще раз
# е - рисовать статистические ошибки в бинах
PAW > hi/plot 20 e
```

При подгонке этого распределения основной интерес представляло его ширина: σ=(4.3±0.3)%. Важно не только значение подгонки, но и оценка ошибки. Например, результат для sigma отличается от того, что должно быть в идеале больше чем на десять ошибок – можно сделать вывод, что есть какая-то, даже не проблема, а «плюха».

Функции

Функции также являются базовым объектом для *PAW*. Для отрисовки одномерных функций используется команда **FUNCTION/PLOT**.

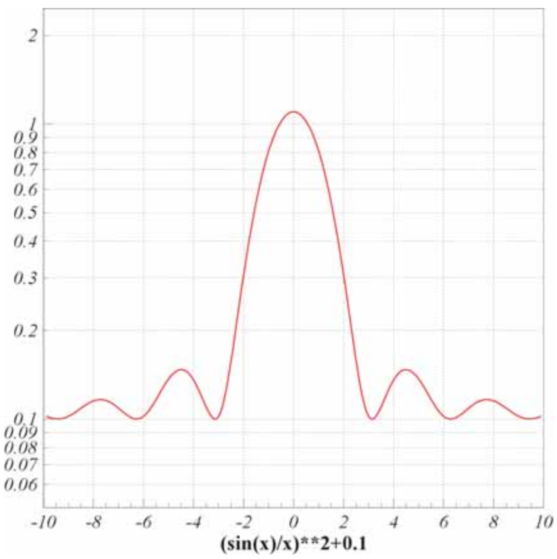
```
#включить логарифмический масштаб для оси Y
PAW > opt logy
#нарисовать одномерную функцию
PAW > fun/plot (sin(x)/x)**2+0.1 -10 10
#вернуться к линейному масштабу для оси Y
PAW > opt liny
```

Обратите внимание на инструкцию *opt* (help GRAPHICS/OPTION). Эта инструкция по своим функциям схожа с командой set, но в отличии от нее отвечает за организацию представления данных, а не за графическое оформление.

Работу с двумерными функциями продемонстрируем на классическом фрактальном изображении имени Мандельброта. Создадим код на FORTRAN:

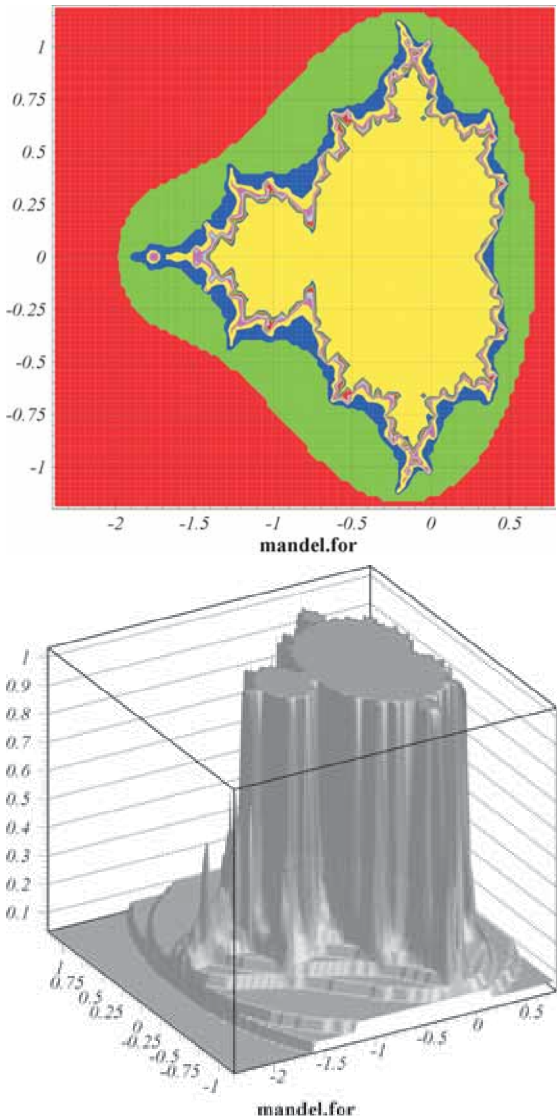
```
C Из официальной документации к PAW
C Файл mandel.for
      real function MANDEL(XP)
      dimension XP(2)
      data NMAX/30/
      x=XP(1)
      y=XP(2)
      xx=0.
      yy=0.
      do n=1,NMAX
         tt=xx*xx-yy*yy+x
         yy=2.*xx*yy+y
         xx=tt
         if (4..lt.xx*xx+yy*yy) goto 1
      enddo
1     MANDEL=FLOAT(n)/FLOAT(NMAX)
      end
```

В случае двумерных функций проблема отображения стоит гораздо острее, чем у одномерных. Двумерные функции для отображения преобразуются в гистограммы (help fun2)



Пример одномерной функции.

```
# По результатам вычисления mandel.for создаем гистограмму 10
PAW > fun2 10 mandel.for 100 -2.4 .8 100 -1.2 1.2 ' '
# Выводим гистограмму 10 как контур
PAW > hi/pl 10 cont3
# Выводим гистограмму 10 как поверхность
PAW > hi/pl 10 surf4
```



Множество Мандельброта. Опции CONT3 и SURF4, соответственно.

Если разрешение не удовлетворяет, то можно создать гистограмму не 100×100, как в примере, а 1000×1000.

Заключение

Объять необъятное совершенно не реально, особенно при лимите на объем текста. Официальная документация содержит около 500 страниц, причем, один алфавитный указатель занимает 17 страниц. *PAW* - матерый программный продукт, которому уже двадцать лет. Этому пакету есть достойный приемник, правда, не лишенный недостатков, но об этом в следующий раз. **LXF**

ЧЕРЕЗ МЕСЯЦ

Когда команде разработчиков PAW стало скучно, они взялись за разработку нового инструмента – ROOT. Вот о нем-то мы и поговорим подробнее.

Программа конференции

Платиновый спонсор: IBM   Соорганизатор конференции: Linux Ink

Конференцзал. Зал №1 ("Синий зал")

4 сентября, понедельник

10:30-10:40 **Приветственное слово:**

ВО РЭСТЕК, Linux Инк, IBM, МЭРТ России, Росинформтехнологии.

Бизнес-секции: Практика использования Linux

10:40-14:45 **Секция №1 Электронное правительство: повышения качества управления за счет решений Open Source**

- **использование решений с открытым кодом в органах государственной власти;**
- **свободные и открытые лицензии, сопряжение с патентным правом;**
- **авторское право.**

Участники: представители Росинформтехнологий; представители Минэкономразвития России; представители Бюро ЮНЕСКО в Москве; Представители Межрегиональной общественной организации в поддержку программы ЮНЕСКО "Информация для всех".

Выступления:

Открытые стандарты для электронного государства. Брауде-Золотарев М., АНО "Экспертно-аналитический центр" .

13:00-13:30 **Перерыв**

Госзакупки программного обеспечения: кто хозяин? Мировые тенденции и российские инициативы. Гребнев Е., ALT Linux;

Почему правительство Великобритании выбрало SE Linux. Логан Ф., IBM Software Group;

Linux в госструктурах. Сафаров М.М., IBM;

Пути обеспечения информационной безопасности автоматизированных систем государственного управления. Маняшин С.М., ОАО "ВНИИНС".

14:45-15:00 **Перерыв**

15:00-17:15 **Секция №2 Средства обеспечения информационной безопасности** Совместный семинар IBM и VDEL "SE Linux: расширенная безопасность в Linux"

Выступления:

Теоретические основы информационной безопасности Linux. Федосеев А., Linux Technology Center, IBM;

Применение SE Linux в проекте для правительства Великобритании. Шанкар К., IBM System&Technology Group;

SE Linux в Red Hat Enterprise Linux - реализация и использование. Меганов А., VDEL;

Сертификация RHEL в России. Болгарчук О., центр компетенции Linux, IBM;

Кофе, вопросы и ответы.

5 сентября, вторник

10:15-12:00 **Бизнес-секции: Практика использования Linux**

Выступления:

Построение корпоративной IT-инфраструктуры на базе Open Source-решений. Дмитриев Д., Linux Инк;

Опыт внедрения решений на базе платформы Red Hat Linux с применением технологий HP, IBM и Oracle. Сивохин Д.В., R-Style;

Программные решения IBM на платформе Linux. Плешек А., IBM;

Легальная миграция на Linux. Курячий Г., ALT Linux;

Продвижение решений на базе платформы Red Hat Linux. Программа Try&Buy. Сивохин Д.В., R-Style.

12:00-12:15 **Перерыв**

Troy Dawson, Fermilab ComputingDivision/CSS CSI Group, разработчик дистрибутива Scientific Linux2.

13:15-13:30 **Перерыв**

13:30-15:00 **Секция №2 Открытые стандарты - путь к надежности и совместимости программных систем**

Выступления:

Обеспечение соблюдения стандартов и проверка надежности Linux с помощью автоматизированного тестирования. Рубанов В., Центр верификации Linux;

Поддержка открытых стандартов корпорацией IBM. Сосновцев Д., IBM;

Zemlin J., FSG;

GO4IT - Платформа тестирования соответствия открытым стандартам Интернета. Пакулин Н., консорциум GO4IT.

15:00-15:15 **Перерыв**

15:15-17:00 **Технические секции: Передовые IT-технологии**

Выступления:

Технология верификации программных интерфейсов. Петренко А.Н., Центр верификации Linux, ИСП РАН;

Realpolitik в LINUX-ONLINE. Опыт разработки и внедрения настольных решений в России и за рубежом. Соколов Е., ООО "Линукс-Онлайн" / LINUX-ONLINE;

Распределенные информационно-вычислительные комплексы (P2P, Grid, поисковые машины, порталы и др.). Садов О., Linux Инк;

МСВС как операционная платформа создания защищенных автоматизированных систем. Жуков И.Ю., Ефанов Д.В., ОАО "ВНИИНС";

Виртуализация управления информационными ресурсами Linux в современной enterprise-системе на базе Active Directory / FDS. Технологическое решение для прозрачной интеграции. Королев Т., ООО "Линукс-Онлайн" / LINUX-ONLINE.

Зал №2

5 сентября, вторник

Секция IBM и Novell "SUSE Linux Enterprise 10: новые возможности"

Совместный семинар IBM и Novell посвящен выходу SUSE Linux Enterprise 10 - новой гибкой и безопасной платформы для открытого предприятия, полностью готовой к поддержке приложений и баз данных, критически важных для бизнеса.

В ходе семинара вы узнаете о новых возможностях, которые вам предлагают серверный дистрибутив SUSE Linux Enterprise Server 10 и дистрибутив для рабочей станции SUSE Linux Enterprise Desktop 10. Отдельно будут отмечены усовершенствования, сделанные сотрудниками IBM, в том числе в плане поддержки оборудования и программного обеспечения IBM. Также будет проведена демонстрация SLES 10 и SLED 10 совместно с программным обеспечением IBM.

Семинары компаний. Зал №4

4 сентября, понедельник

11:00-12:30 **Пресс-конференция по выставкам Infosecurity, Storage Expo, Documation**

13:30-14:30 **Семинар компании "Лаборатория Касперского"** "Kaspersky Hosted Security - новая услуга аутсоринга защиты корпоративной почты"

15:00-16:30 **Пресс-конференция компании "Доктор Веб"** "Современные компьютерные угрозы и уникальные технологии для борьбы с ними. Тенденции и прогнозы"

5 сентября, вторник

11:00-13:30 **Семинар компании IBM** "Системы хранения данных: решения и технологии IBM"

14:00-17:00 **Семинар компании "Открытые технологии"** "Решения по обеспечению информационной безопасности критичных бизнес-приложений"

6 сентября, среда

10:30-11:30 **Семинар компании "Доктор Веб"** "Построение эффективной системы защиты информационных ресурсов предприятия. Антивирусные решения Dr.Web"

12:00-14:00 **Семиар компании "Инфосистемы Джет"**

Программа конференции предварительная. Возможны изменения.

ПРИГЛАСИТЕЛЬНЫЙ БИЛЕТ

Навигатор по ДИСТРИБУТИВАМ LINUX

Mandriva Linux

Дистрибутив Mandriva (ранее, до слияния с Connectiva называвшийся Mandrake), по праву считается идеальным для неподготовленного пользователя. Mandriva Linux невероятно прост в установке благодаря современному графическому установщику. Гордость дистрибутива – комплект графических утилит для настройки системы, объединенные в Mandriva Control Center. Установив Mandriva, вы сможете настроить абсолютно все, от звуковой карты до файервола, не прибегая к помощи командной строки.

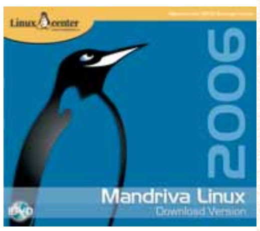
Разработчики Mandriva Linux очень тщательно подходят к тестированию программных пакетов, поэтому в новые версии дистрибутива выходят сравнительно редко, но зато содержат только стабильные программы.

Mandriva One – однодисковая версия дистрибутива, которая прекрасно подойдет для первого знакомства с системой. Mandriva One представляет собой LiveCD с возможностью установки на жесткий диск.



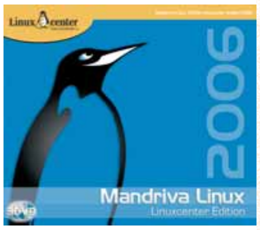
Варианты поставки: 1CD

Mandriva Linux 2006 Download Edition – полностью свободная версия дистрибутива, не содержащая проприетарных (закрытых) компонентов и доступная для бесплатной загрузки.



Варианты поставки: 3CD или 1DVD

Mandriva Linux 2006 LinuxCenter Edition – дистрибутив на основе Download Edition, собранный специалистами ЛинуксЦентра. Отличается от Download Edition большим набором пакетов. Подойдет тем, кто не имеет возможности скачивать пакеты из Интернета и хотел бы иметь все необходимое на компакт-дисках.



Варианты поставки: 1DVD или 3DVD (в зависимости от количества пакетов)

Mandriva Linux 2006 Discovery/LX – это комплект, содержащий LiveCD MandrivaMove, установочную версию с включенными Real Player, Flash Player и Adobe Reader, а также месяц бесплатной технической поддержки и серебряную клубную карту MandivaClub.



Варианты поставки: 4CD, печатная документация на русском языке.

Mandriva Linux 2006 PowerPack – это полноценная настольная ОС, которая будет хорошо смотреться на офисном ПК или рабочем месте разработчика. PowerPack придется по душе опытным пользователям Linux, поскольку поддерживает новую технологию виртуализации Xen.



Варианты поставки: 2DVD (1DVD с 32-битной и 1DVD с 64-битной версией), печатная документация на русском языке

Mandriva Linux PowerPack+ удачно сочетает в себе мощь Mandriva Linux 2006 PowerPack с первоклассными серверными приложениями. Это – наиболее полная редакция в линейке Mandriva Linux. Благодаря своей простоте в установке и настройке, PowerPack+ хорошо подходит для сегмента SOHO.



Варианты поставки: 2DVD (1DVD с 32-битной и 1DVD с 64-битной версией), 7CD, печатная документация на русском языке.

SUSE Linux

SUSE Linux считается самым популярным дистрибутивом в Европе. Первоначально разрабатывался компанией S.u.S.E. Linux GMBH, ныне купленной Novell. SUSE Linux совместим с большим количеством оборудования, включая нестандартное, причем большинство устройств настраивать не придется – обычно все автоматически конфигурируется во время установки.

Отдельно хотелось бы отметить комплекс конфигурационных утилит YaST, содержащий все, что нужно для настройки системы (и даже больше). SUSE Linux всегда отличалась стабильностью и качественно собранными приложениями – если в какой-то программе, входящей в дистрибутив, обнаруживается ошибка, разработчики не включают ее в систему до тех пор, пока она не будет исправлена.

Все версии SUSE Linux (кроме Enterprise Server) прекрасно подойдут начинающим пользователям, не имеющим опыта работы с Linux.

OpenSUSE Linux 10.1 (Goldmaster) – это свободно распространяемая версия дистрибутива, разрабатываемая сообществом. На основе пакетов, собранных сообществом для OpenSUSE, Novell готовит коробочные версии дистрибутива и корпоративные продукты.



Варианты поставки: 5CD или 1DVD (доступны версии для платформ x86 и x86_64)

SUSE Linux 10.1 – коробочная версия дистрибутива, включающая в себя все необходимое для офисной работы, развлечений, мультимедиа, Интернета. В эту версию дистрибутива входят несвободные компоненты, такие как антивирус Antivir PE, Real Player, Main Actor.



Варианты поставки: коробка с CD- и DVD-версией дистрибутива (для платформ x86 и x86_64) и печатная документация

SUSE Linux Enterprise Desktop – специальная версия дистрибутива SUSE, созданная для корпоративных нужд. Целью Novell было создание системы, которая может заменить Windows в офисе, при этом разработчики SLED постарались сделать так, чтобы переход с Windows был простым для администратора и безболезненным для пользователя. SLED может интегрироваться в гетерогенные сети, поддерживает Active Directory, eDirectory и другие службы каталогов. С помощью AutoYaST можно создать сценарий автоматической установки, что упрощает развертывание системы. Пользователь в свою очередь получает современный рабочий стол Gnome с переработанным главным меню и поддержкой спецэффектов Xgl.



Варианты поставки: 5CD или 1DVD (доступны версии для платформ x86 и x86_64)

SUSE LINUX Enterprise Server 10 (SLES) представляет собой масштабируемую, высокопроизводительную платформу безопасных корпоративных вычислений, реализующая все преимущества Linux и Open-Source. Система ориентирована на сервера для ответственных корпоративных приложений и обеспечивает высочайшую надежность, производительность и функциональность.



Варианты поставки: 4CD или 1DVD (доступны версии для платформ x86 и x86_64)

Red Hat

После 2003 года Red Hat занимается исключительно системами корпоративного уровня. Red Hat Linux всегда был, и по сей день остается тем стандартом, на который опираются производители коммерческих приложений и серверов. Дистрибутив сертифицирован Oracle. Самым ценным в Red Hat является поддержка – время реакции не превышает 4 бизнес-часа, поддержка доступна как через web-интерфейс, так и по телефону. Кроме того, покупателям продуктов Red Hat доступны самые свежие обновления в кратчайшие сроки, через систему Red Hat Network.

Red Hat Enterprise Linux WS – многофункциональная системы для рабочих станций, подходящая для разработки программного обеспечения, CAD-приложений и решения других технических ресурсоемких задач. Система может работать как на одно-, так и на двухпроцессорных рабочих станциях, что позволяет в полной мере задействовать возможности имеющегося оборудования.



Варианты поставки: коробка с CD-версией для платформ x86 и x86_64

Red Hat Enterprise Linux 4 ES ориентирован на использование в качестве сервера отдельного департамента или небольшой организации. С его помощью можно легко развернуть web-сервер, службу печати или файловый сервер. Red Hat Enterprise Linux 4 ES может работать на системах, имеющих до двух процессоров и до 16Гб основной памяти.



Варианты поставки: коробка с CD-версией для платформ x86 и x86_64

Red Hat Enterprise Linux 4 Advanced Server (AS) – прекрасная платформа для поддержки баз данных, корпоративных приложений ERP/CRM и решения бизнес-критичных задач. Это единственная редакция Red Hat Enterprise Linux, способная работать на высокопроизводительных серверах серии IBM Power и мейнфреймах IBM S/390 или zSeries. Red Hat Enterprise Linux 4 AS не имеет ограничений на объем оперативной памяти и число процессоров.



Варианты поставки: коробка с CD-версией для платформ x86 и x86_64

Fedora Core

Fedora Core – сообщество энтузиастов, создающих свободную и надежную основу для дистрибутивов Red Hat: каждое нововведение в Red Hat Enterprise Linux сначала появляется в репозитории Fedora Core. Каждый релиз Fedora Core содержит самые новые версии программ, поэтому этот дистрибутив подойдет энтузиастам и любителям экспериментов. У Fedora Core огромное сообщество пользователей: если вы встретитесь с какими-либо проблемами в ходе использования дистрибутива, будьте уверены – на форумах и списках рассылки вам обязательно помогут.

Fedora Core 5 прекрасно подойдет для использования на десктопе, однако справится с задачами небольшого сервера. За использованиеFedora на сервере можно приводить множество аргументов, но самым весомым будет скорость выпуска критичных обновлений и цикл поддержки дистрибутива. Как только очередная версия дистрибутива перестает поддерживаться разработчиками, ее «подхватывает» проект Fedore Legacy, предоставляющий критичные обновления для входящих в дистрибутив пакетов.



Варианты поставки: 5CD или 1DVD (доступны версии для платформ x86 и x86_64)

Fedora Core 5 UPDATES – комплект критичных обновлений для дистрибутива Fedora Core 5.



Варианты поставки: 1DVD

Fedora Core 5 Extras – комплект дополнительных пакетов для дистрибутива Fedora Core 5, выпускаемых проектом Fedora Extras. Здесь вы найдете все те, программы, которые не вошли в дистрибутив.



Варианты поставки: 1DVD

Linux XP

Linux XP 2006 — это настольный Linux на базе передовых технологий RedHat и Novell. Разработчик системы, компания Linux-Online, известен в России и в мире своим нестандартным видением в области продвижения Linux среди начинающих пользователей. Linux XP 2006 предназначен для начинающих пользователей Linux, которые ценят свое время и стремятся к удобствам готового и простого решения. Тщательная русификация, оригинальный дружелюбный интерфейс, простота установки и управления, продуманность набора приложений, ряд уникальных функций – все это создает непревзойденный комфорт для пользователя.

Linux XP 2006 SR1 – система, совместимая с Red Hat и Fedora Core, предоставляющая пользователю основанный на Gnome рабочий стол, который разработчики слегка модифицировали, чтобы сделать его похожим на рабочий стол Windows. Базовая система Linux XP содержит по одному приложению для каждой задачи. Linux XP 2006 SR1 включает в себя офисный пакет, средства разработки, игры и «домашний кинотеатр» на LiveCD.





Варианты поставки: 5CD

Linux XP 2006 Ultra Edition SR1 представляет собой обычный Linux XP 2006 с обновлениями и расширенным набором пакетов, которые не вошли в стандартную версию.
Варианты поставки: 2DVD

Knoppix

Knoppix – один из наиболее популярных дистрибутивов Linux в формате LiveCD/LiveDVD (то есть запускающийся прямо с диска и не требующий установки), хорошо подходящий как для ознакомления с возможностями Linux, так и для тестирования оборудования и организации мобильного рабочего места. KNOPPIX основан на дистрибутиве Debian GNU/Linux Sid. KNOPPIX -- универсальный LiveCD/LiveDVD, который подойдет каждому: и системному администратору, которому нужно срочно восстановить загрузчик на сервере, и менеджеру, которому срочно нужно напечатать отчет (а Windows как раз пал жертвой вируса), и простому домашнему пользователю, который только присматривается к Linux.

Knoppix 5.0.1 – оригинальная версия Knoppix, собранная Клаусом Кноппером. Использование файловой системы UnionFS позволяет в течение рабочего сеанса устанавливать дополнительные приложения с помощью мощного менеджера пакетов apt. Кроме того, KNOPPIX можно установить на жесткий диск, таким образом, KNOPPIX -- это еще и самый быстрый способ развертывания Debian GNU/Linux.



Варианты поставки: 1CD или 1DVD (в зависимости от количества пакетов)

Knoppix 5 Russian Edition – это русская редакция оригинального Knoppix, собранная специалистами ЛинуксЦентра. В эту версию внесено много улучшений и исправлений, а также включена поддержка русского языка во всех программах первой необходимости. Русифицированы OpenOffice, Mozilla Firefox, Mozilla Thunderbird, KDE, а также системные утилиты Knoppix.
Варианты поставки: 1CD

Alt Linux

Alt Linux – универсальный отечественный дистрибутив, разрабатываемый энтузиастами со всего СНГ. Каждая версия Alt Linux основана на репозитории пакетов Sisyphus, который содержит тысячи пакетов. В Sisyphus содержится большое количество довольно редких программ, которые вы вряд ли найдете в других дистрибутивах.

Alt Linux 3.0 Compact, Travel CD запускается непосредственно с компакт-диска. Вы можете полноценно использовать компьютер даже в том случае, если в нем нет жесткого диска. Travel CD основан на тех же программах, что и основная версия дистрибутива: интернет-браузеры и почтовые клиенты, офисные приложения, приложения для просмотра и прослушивания мультимедиа. Travel CD прекрасно подходит для проверки совместимости компьютерного оборудования с операционной системой Linux.



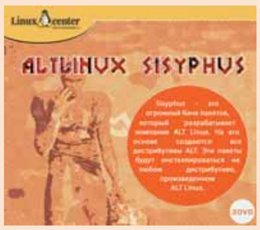
Варианты поставки: 1CD

Alt Linux 3.0 Compact – простая и удобная операционная система для повседневной работы, основанная на последних разработках ALT Linux. Дистрибутив содержит всё необходимое для пользователя домашнего компьютера, в том числе программы для работы в Интернет, офисные приложения, программы для просмотра и прослушивания мультимедиа. В Compact реализована поддержка широкого спектра современного оборудования, в том числе процессоров, материнских плат, видеоускорителей, принтеров, цифровых камер, сканеров.



Варианты поставки: 1CD или 1DVD (в зависимости от количества пакетов)

Alt Linux Sisyphus -- это огромный банк пакетов (репозиторий), который разрабатывает компания ALT Linux. На его основе создаются все дистрибутивы ALT. Поддерживаемая ALT Linux Team целостность Sisyphus, оригинальная технология сборки пакетов, интегрированная в дистрибутивы утилита apt-get и ее оболочки aptitude и synaptic позволяют пользователям легко обновлять свои системы и быть в курсе всех новостей мира свободных программ. Новые версии выпускаются раз в месяц.



Варианты поставки: 2DVD

ASP Linux

ASP Linux – это отечественный дистрибутив, основанный на Fedora Core. В отличие от Fedora Core, ASP Linux пригоден к использованию сразу после установки, и практически не требует каких-либо дополнительных усилий по настройке. В дистрибутиве уже проведена грамотная русификация, установлены видео-драйверы с поддержкой аппаратного ускорения, включены аудио- и видео-кодеки. ASP Linux прекрасно подойдет для первого знакомства с Linux. В дистрибутиве используется свой установщик, который позволяет изменять размер разделов (в том числе и с файловой системой NTFS), а также свой загрузчик.

ASPLinux 11 Greenhorn – реализация принципа LiveCD на основе новой версии ASPLinux 11. Этот вариант дистрибутива не требует установки на жесткий диск и запускается прямо с CD.



Варианты поставки: 1CD

ASPLinux 11 Express – дистрибутив ASPLinux 11, в котором нет ничего лишнего. Квалифицированные пользователи Linux не всегда нуждаются в печатных руководствах по установке и настройке системы и, как правило, хорошо представляют, какие именно приложения и пакеты они собираются устанавливать и использовать.



Варианты поставки: 4CD

ASPLinux 11 Standard – типовой вариант дистрибутива ASPLinux 11 для установки на компьютеры корпоративных или домашних пользователей. Этот вариант дистрибутиа – лучший выбор для пользователей, которые хотели бы работать в Linux, но не обладают навыками программиста или системного администратора. В комплект также включены игры и демоверсии коммерческих приложений под Linux.



Варианты поставки: коробка с 6CD и двумя книгами

ASPLinux v10 Deluxe – наиболее полный вариант дистрибутива ASPLinux v10. Это идеальный помощник специалиста, которому необходим надежный и мощный инструмент для решения любой задачи, какой бы сложной она не являлась. Этот дистрибутив с одинаковым успехом может быть использован для установки как на рабочей станции, так и на сервере предприятия малого или среднего бизнеса. В эту версию включены диски с исходными текстами, документацией, играми и Acronis OS Selector.



Варианты поставки: коробка с 10CD, 1DVD и тремя книгами

ASPLinux Server II оптимизирован для создания корпоративных серверов различных классов и поддерживает различные серверные архитектуры, включая многопроцессорность и большие объемы памяти. ASPLinux Server II включает все необходимые средства для построения, настройки и администрирования почтового сервера, Интернет- и веб-сервера, сервера печати, сервера приложений, сервера баз данных, файлового сервера.



Варианты поставки: коробка с 8CD, двумя книгами и футболкой.

Вопрос? Ответ!

Если вы завязли в какой-то проблеме и чтение HOWTO не помогает, почему бы не написать нам? Наши эксперты помогут разобраться даже в самых сложных проблемах.

НАШИ ЭКСПЕРТЫ

Наши эксперты найдут ответ на самый трудный ваш вопрос. Если у вас проблемы с установкой, настройкой модема, сетью или еще чем-нибудь – просто напишите нам, а обо всем остальном позаботимся мы.



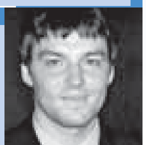
Управляя Интернет-провайдером, а заодно подрабатывая редактором дисков *LXF*, **Нейл Ботвик** (Neil Bothwick) скромно зовет себя мастером на все руки.



Брэндон Калигари (Brendon Caligari) больше десяти лет работает с Linux, он администратор работающей системы в фирме Rackspace Managed Hosting.



Александр К. – сторонник Unix-way. Молодой, но перспективный член дружной команды экспертов.



Валентин Синицын Поддерживает проект Slackware Reiser4, интересуется настольными Linux-технологиями и рад помочь Вам разобраться с ними.



NFS и Mepis

В Мне понравилась версия Mepis из *LXF79*, с ней у меня даже впервые нормально заработал Skype под Linux. Однако, к моему разочарованию, не нашлось NFS, и я не могу пользоваться своей Linux-сетью. Сайт Mepis не помог, у других те же проблемы. Пришлось вернуться на Kubuntu, который довольно похож и всё хорошо делает (хотя, к сожалению, *Midnight Commander* недоступен в обеих системах). Можете ли вы дать мне инструкции, как заставить работать NFS в Mepis?

Дж.Ф.Л. [JFL]

О Хотя в ядре Mepis включена поддержка NFS, в самом дистрибутиве нет *Portmap*, который нужен для монтирования NFS-разделов. Запустите *Synaptic*, перейдите в Настройки > Репозитарии (Settings > Repositories) и отметьте первый репозитарий, Debian. Щелкните по кнопке «Получить сведения» (Reload), чтобы обновить списки пакетов, а затем, воспользовавшись кнопкой Искать (Search), найдите и установите *Portmap*. Вам также нужно проверить, что сервис *Portmap* запущен, когда Вы загружаетесь – установка должна об этом позаботиться.

Теперь Вы можете смонтировать NFS-ресурс с помощью стандартной команды:

```
mount -t nfs hostname:/exported/dir /mnt/somewhere
```

Если Вы пытались смонтировать ресурс в этой же сессии, и у Вас ничего не вышло, может быть, надо просто перезагрузиться. NFS иногда капризничает.



FNFS и *Midnight Commander* будут доступны в Mepis после добавления репозитория Debian.

Midnight Commander станет доступен для Mepis, когда Вы активируете репозиторий Debian, необходимый для установки *Portmap*. Вы найдете его, пошарив в *Synaptic*. Пакет называется **mc**. ***НБ***

- По поводу *Skype* смотрите «Краткую справку» на следующей странице.

В незнании – благо

В В нашей интранет-сети около полутора тысячи компьютеров под Windows 2000 с доступом к нашему web-серверу под Debian Sarge и *Apache 2.0.54*. Похоже, на некоторых из них запущен сервис WebDav, подключающийся к интранет-серверу и забивающий информацией мои журналы.

Есть ли способ заставить Apache просто не знать о запросах, сделанных этим сервисом? Его User-agent –'Microsoft-WebDAV-MiniRedir/5.1.2600'.

Длanни, с форума LXF [Dlunny]

О Вы можете блокировать (или разрешить) запросы с определенных агентов с помощью комбинации из директив **SetEnvIf** и **Deny** (или **Allow**), которые можно включить в секцию <**Directory**> конфигурационных файлов **httpd.conf** или **.htaccess**. Так как Вы хотите блокировать все запросы с этого агента, я бы посоветовал добавлять директивы в секцию **<Directory>** каталога **DocumentRoot**.

Директивы для блокирования выглядят так:

```
SetEnvIf User-Agent ^Microsoft-WebDAV-MiniRedir BegoneWebDAV
Order Allow,Deny
Deny from env=BegoneWebDAV
```

Первая строка настраивает переменную окружения **BegoneWebDav** для агентов, имена которых начинаются с 'Microsoft-WebDAV-MiniRedir', так что она будет продолжать работать, даже если номер версии сервиса изменится. Следующая переменная, будучи установленной, блокирует доступ. Комбинация **SetEnvIf**, **Allow** и **Deny** дает отличную возможность для контроля, кто или что может иметь доступ к определенным частям Вашего ресурса. Для дополнительной информации ознакомьтесь со следующими документами:

- **httpd.apache.org/docs/2.0/mod/mod_setenvif.html#setenvif**
- **httpd.apache.org/docs/2.0/mod/mod_access.html#deny**
- **httpd.apache.org/docs/2.0/mod/mod_access.html#allow**

НБ

Хочется радушия

В Как можно настроить приглашение к авторизации? Хочу, чтобы оно выглядело примерно так, как на сайте **http://alinux.org/linux-os/aLinux-step4.png**, но без графики и статичного текста. Я хочу пингвина, информацию о ЦПУ, памяти, производительность процессора и приглашение. Вы можете помочь?

Майк86, с форума LXF [Mike86]

О Текст, выводящийся на терминал перед приглашением к авторизации, берется из файла **/etc/issue**. Поместите то, что Вы хотите видеть, включая ANSI-графику, в этот файл. В него можно даже добавлять, при помощи *Cron*, текущие напоминания вроде «Следующее воскресенье – день матери» или «Купить свежий *Linux Format* завтра».

Если Ваших художественных способностей маловато для создания ANSI-пингвина, поможет пакет *Linux_logo*, доступный на **www.deater.net/weave/vmwprod/linux_logo** и, возможно, в репозитариях Вашего дистрибутива. Man-страница для *Linux_logo* описывает множество опций для управления выводом. Приведённый Вами пример создан с помощью такой команды:

```
linux_logo -c -y -k >/etc/issue
```

Удачи! Окружающая текст графика – это фрейм-буфер, не имеющий отношения к приглашению авторизации. ***НБ***

Пропавший профиль

В Я бы хотел создать резервные копии моих закладок, настроек и адресной книги *Firefox* и *Thunderbird*. В Windows у меня была для этого отдельная программа. Но в Fedora я не смог найти свой профиль. Пути,



Создайте из *Linux-logo* свою графику ANSI.

предлагаемые на сайте *Mozilla*, не похожи на место хранения моих закладок (хотя я нашел закладки Red Hat) или моей почты. Поиск по файловой системе ничего не дал. Видимо, я задавал системе неправильные вопросы...

Джон Х. Браун [John H Brown]

О **Firefox хранит свои настройки в ~/.mozilla/firefox/default.???**, где **???** – случайная строка. *Thunderbird* использует **~/.thunderbird/default.???**. Например, мои закладки, настройки и адресная книга хранятся соответственно в

- **~/.mozilla/firefox/default.yyh/bookmarks.html**
- **~/.thunderbird/default.piz/prefs.js**
- **~/.thunderbird/default.piz/abook.mab**

Вероятно, простейший способ сделать резервные копии полных директорий такой:

```
tar czf FfandTBsettings.tar.gz ~/.mozilla/firefox ~/.thunderbird
```

Вы можете воспользоваться *Cron* для автоматического ежедневного создания резервных копий, сохранив следующий скрипт в **/etc/cron.daily/mozbackup**:

```
#!/bin/sh
tar czf /somewhere/safe/FFandTBsettings-$(date -I).tar.gz /home/john/.mozilla/firefox /home/john/.thunderbird
```

Не забудьте установить его бит исполняемости, иначе он не запустится:

```
chmod +x /etc/Cron.daily/mozbackup
```

НБ

Магия бессильна

В Я купил ваш спецвыпуск про *Fedora Core 5*, и следовал инструкциям по установке, но наткнулся на проблемы. У меня установлена Windows XP, и с помощью *Partition Magic* я создал Linux-раздел на 10 ГБ.

Установка с диска шла хорошо до шестого этапа, после чего установщик принялся сообщать о невозможности продолжения. Я предупреждал *Partition Magic*, что хочу установить Linux – может, мне надо разбить этот раздел на три части, чтоб



Оставьте свободное пространство на жестком диске неразмеченным, установщик Fedora Core создаст разделы сам..

установить три директории?

Тим [Tim]

О Linux-разделы, созданные с помощью *Partition Magic*, иногда создают проблемы. В установочных программах многих дистрибутивов есть опция изменения размера Windows-раздела и создания Linux-разделов, предпочтительно ею и пользоваться. Fedora такую возможность не предоставляет, так что лучше будет с помощью *Partition Magic* изменить размер раздела Windows, но пространство Linux-раздела оставить неразмеченным.

А сейчас – удалите созданные Вами Linux-раздел(ы). После этого запустите установщик Fedora Core 5 и выберите «Использовать свободное пространство». Установщик создаст нужные разделы сам на свободном месте. Важно помнить, что под свободным местом понимается неразмеченное пространство, а не место в готовых разделах. ***НБ***

КРАТКАЯ СПРАВКА: VOIP

VoIP (Voice over IP) – это использование TCP/IP-соединения для двунаправленной передачи голоса. Многие имеют дешевое высокоскоростное Интернет-соединение, а при телефонных звонках надо платить и за время разговора, и за дистанцию. VoIP приобретает все большую популярность в офисной среде: для телефонии и Интернета используется одна и та же инфраструктура (кабель и пр.), и обычный компьютер может служить коммутатором, сервером голосовой почты и шлюзом во внешний мир, используя VoIP-связи или даже обычные телефонные линии респондентов.



***Skype* – бесплатная программа для звонков по всему миру**

Идея не нова, однако сдерживающими факторами всегда были скорость соединения и его качество. Другая сложность – обеспечение связи через межсетевые фильтры. Однако *Skype* (**www.skype.com**) смог решить эти проблемы, создав приемлемое качество связи на хорошем коммутируемом соединении, а при использовании более высокоскоростного канала – близкое к телефонному. Он также может работать через стандартные web-порты 80 и 443 – многие брандмауэры их не запирают. Однако главная причина популярности *Skype* – тот факт, что он «просто работает» на множестве ОС (Linux x86, Windows, MacOS).

Есть открытые альтернативы Skype, использующие протокол SIP, например, KPhone и Linphone для Linux и SJPhone для Windows. Все они требуют регистрации у SIP-провайдера. Единожды подключившись, вы можете позвонить любому человеку, использующему SIP, даже если у него другой SIP-провайдер. Список провайдеров имеется на **www.sipcenter.com/sip.nsf/html/Service+Providers**.

Разговоры с другими пользователями *Skype* бесплатны, и у большинства SIP-провайдеров тоже. При звонках на стационарные телефоны VoIP дает заметную экономию. Поскольку трафик передается через Интернет до ближайшего к абоненту узла, звонки на другой континент обходятся по тарифу местного вызова.

Пакости распаковки

В Три недели назад я установил SUSE 10.0, и с тех пор мне незачем оглядываться. Надеюсь со временем полностью перейти на Linux. Одна проблема, не могу разобраться с установкой программ из архивов. Я так понял, их надо распаковать и перейти в их директорию, однако когда ввожу **./configure**, получаю вот что:

```
bash: ./configure: No such file or directory
```

Я немного озадачен: всё делал по советам многих людей, пытался выполнить это как суперпользователь, но до сих пор ничего не выходит. Чувствую, я упустил что-то очень простое.

Джи М. Николсон [Gee M Nicholson]

О Часто говорят, что запуск **./configure** является первом шагом после распаковки архива, а на самом деле он третий. Первые два:

1) Найдите файлы с инструкциями по установке.

2) Внимательно прочитайте их.

Большинство архивов с исходными кодами содержат файлы **README** и **INSTALL**, которые надо просмотреть и понять, как установить программу. Стандартный метод установки программ из исходных кодов

```
./configure
make
make install
```

подходит в более чем 90% случаев, но есть и исключения. Иногда настройка не нужна – в частности, для очень простых программ. То есть Вы должны выполнить только **make** и **make install**. Бывает, что программа использует иной способ установки. В любом случае Вы должны предварительно ознакомиться с инструкциями.

Несмотря на нежелательность исполнения первых двух шагов от имени суперпользователя, его прав обычно требует **make install** для записи файлов в системные директории. Так как это потенциальная опасность, прочитайте инструкции обязательно.

Другой полезный шаг при использовании *./configure* – запустить его сначала с параметром **--help**. Это даст Вам возможность контролировать сборку и установку программы (другие способы для этого отнюдь не просты). ***НБ***

Управлять движением

В Моя система будет служить шлюзом в Интернет, а также файловым сервером для моей локальной сети, в ней две Ethernet-карты. Я собираюсь настроить *Iptables*; есть ли способ ограничить сервисы (*Samba*, NFS и т.д.), чтобы они работали только с одним

>>

ЧАСТО ЗАДАВАЕМЫЕ ВОПРОСЫ: WINE

FAQ Что такое Wine?

Это рекурсивный акроним: Wine Is Not Emulator.

FAQ Не особо полегчало!

Ну, на самом деле это такой вид эмулятора. Wine – открытая реализация программного интерфейса Windows (WinAPI), работающая поверх X и Unix.

FAQ Wine позволяет запускать Windows из-под Linux?

Нет. Wine – некая прослойка между Windows-программами и Linux. Когда вы запускаете Windows-программу в Wine, она думает, что работает в родной среде, однако ее обращения к Windows-функциям транслируются в вызовы Linux-функций.

FAQ Надо ли для работы в Wine ставить Windows?

Нет, но Wine будет пользоваться DLL-библиотеками из существующей установки Windows, если она у вас есть. Если нет, он будет работать со встроенными альтернативными библиотеками.

FAQ Windows у меня уже есть. Зачем мне Wine?

Вы можете запускать Windows-приложения без перезагрузки, то есть можно одновременно использовать программы и под Windows, и под Linux. Wine также позволяет использовать подключаемые модули Windows [plugins] в Linux-программах, например, web-браузерах и плейерах. Это дает Linux-программам доступ к проприетарным форматам.

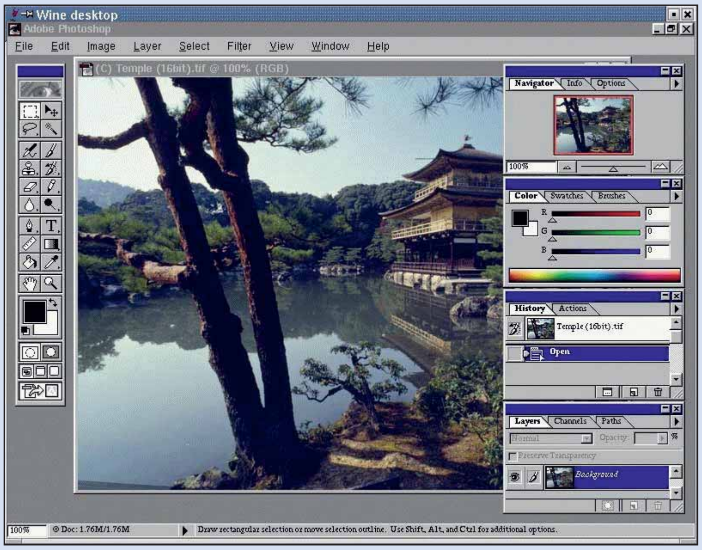
FAQ Эмуляторы часто тормозят. А Wine?

Он эмулирует только API, а не железо, поэтому программы работают с нормальной скоростью, а иногда и быстрее, чем в Windows; но поэтому Wine может быть запущен только на том же оборудовании, что и Windows.

FAQ Где его взять?

Вы можете скачать Wine в различных форматах с www.winehq.com или поискать пакеты в репозиториях к вашему дистрибутиву. Есть и коммерческие варианты Wine. CrossOver Office от Codeweavers позволяет запускать офисные приложения для Windows, а также подключаемые модули к Internet Explorer в нескольких Linux-браузерах, причём тяжкий труд конфигурирования выполняется автоматически. На сайте Codeweavers перечислено около 50 поддерживаемых приложений, но на самом деле их гораздо больше.

С Wine можно работать в Windows-приложениях, например, в Photoshop.



FAQ Как насчёт игр?

Другой коммерческий вариант Wine – Cedega – улучшенная версия с поддержкой DirectX. Она бесплатна через CVS. Если вы хотите получить готовые пакеты и поддержку, понадобится платная подписка.

<< интерфейсом, а именно, моей локальной сетью?

Это нужно делать для каждого сервиса отдельно, или есть способ глобальной настройки? Я буду использовать Fedora Core или SUSE.

Jellyman_4eva, с форума LXF

Для этого есть три способа. Первый состоит в настройке каждого сервиса на работу только с одним сетевым интерфейсом. Если сервисов у Вас немного, этот способ может оказаться самым простым и гибким. Просмотрите man-страницы для каждого сервиса и добавьте подходящие опции в конфигурационные файлы. Пусть IP-адрес Вашего LAN-интерфейса 192.168.0.1, а у другого интерфейса – другой адрес; тогда Вы должны сделать следующее:

- добавить Listen 192.168.0.1:631 в **/etc/cups/cupsd.conf**
- добавить socket address =192.168.0.1 в **/etc/samba/smb.conf**
- добавить Listen 192.168.0.1:80 в **/etc/apache2/httpd.conf** (местоположение файла может быть другим).



NFS слегка отличается: ему нужно для каждого ресурса указать диапазон адресов, которым разрешено соединяться, в файле **/etc/exports**:

```
/path/to/export 192.168.0.0/24(rw,sync)
```

Второй способ – использовать *Iptables* для полной блокировки доступа из глобальной сети к портам нужных сервисов. Вы можете сделать это для каждого порта в отдельности, а заодно и добавить блокировку по первому способу. Альтернатива – блокировать вообще весь входящий трафик, это, кстати, настройка по умолчанию для большинства брандмауэров в Linux. Тогда Вам останется лишь открыть порты для нужных сервисов, вроде SSH. Если Вы не слишком знакомы с *Iptables*, ручное редактирование правил может пробить серьезную брешь в безопасности системы. Поэтому эксперты рекомендуют использовать графические утилиты настройки, например, *Guarddog* или *Shorewall*. И в Fedora Core, и в SUSE есть удобные средства настройки правил брандмауэра.

Третий способ – блокировать доступ к сервисам на Вашем модеме или роутере. Это самый безопасный метод, поскольку Вы останавливаете трафик еще до того, как он попадет на Ваш компьютер. Однако это не всегда возможно – все зависит от Вашего роутера или модема.

Способы не являются взаимоисключающими, Вы спокойно можете комбинировать их для обеспечения большей безопасности – так сказать, «поддержка штанов и ремнём, и подтяжками». ***НБ***

Способ Samba

У меня есть Linux-машина на 192.168.1.1, соединенная с беспроводным роутером. На ней содержится резервная копия данных моего Windows XP-ноутбука, я её туда загружаю по FTP. Нет ли способа получше? Могу ли я хранить свои документы на Linux-машине и получать их оттуда через сеть?

Ричард Уоткинс [Richard Watkins]

Простое решение – используя *Samba*, настроить CIFS-сервер на Linux-машине. Установите последнюю версию *Samba* (**www.samba.org**) и найдите конфигурационный файл: обычно это **/etc/samba/smb.conf**.

Файл конфигурации разделен на две секции: глобальные настройки и установки разделяемых ресурсов. Глобальные настройки относятся к работе самого CIFS-сервера и используются для контроля всего – от сетевого интерфейса, через который работает сервер, до настроек *Active Directory* в Windows. В Вашем случае глобальные настройки можно не трогать.

Теперь настроим Ваш ресурс. Допустим, файлы располагаются в директории **/export/share**, именем ресурса в CIFS будет myshare, описанием – all my files. Теперь, так как в беспроводной сети множество пользователей, надо ограничить доступ к ресурсу, предоставив права на чтение и запись только пользователям Fred и Mary. Добавьте следующие строки в **smb.conf**:

```
[myshare]
comment = all my files
path = /export/share
valid users = mary fred
public = no
```

>>

<<

```
writable = yes
printable = no
create mask = 0765
```

Главное позади, но нужно ещё добавить пароли для Фреда и Мэри. Для этого запускайте *smbpasswd* от имени суперпользователя:

```
# smbpasswd -a fred
New SMB password:
# smbpasswd -a mary
New SMB password:
#
```

Наконец, проверьте, запущена ли *Samba* – если нет, запустите. Теперь Вы можете получить доступ к CIFS-ресурсу с Вашего ноутбука через \\192.168.1.1\myshare. ***КК***

Монтирование LVM

В Можно ли смонтировать *LVM*-разделы внешнего жесткого диска? Я хотел бы скопировать один файл с моего старого жесткого диска с установленной Fedora Core 3, избежав загрузки с него.

Зарфати [Zarfati], с форума LXF

О Если у Вас установлены утилиты *LVM*, Вы можете монтировать *LVM*-разделы с любого диска (мне случалось монтировать *LVM*-раздел, находящийся на флэш-брелке). Запустите от имени суперпользователя

```
vgscan
vgchange -a y
```

После этого все обнаруженные разделы будут видны в системе как устройства **/dev/volumegroup/logicalvolume**, а значит, Вы сможете их смонтировать обычным способом:

```
mount /dev/volumegroup/
logicalvolume /mnt/somewhere
```

НБ

Аварийная мигалка

В У меня на компьютере с двумя жесткими дисками установлена Mandriva 2006. Второй диск отведен под резервные копии. После загрузки KDE всё отлично, только индикатор жесткого диска принимается беспрерывно мигать – да так, что я и сосчитать не успеваю. То же самое – на старой машине, где я тестирую всякие программы.

Мне присоветовали удалить *Kat*, но это не помогло. Я попробовал на старой машине SUSE, и всё стало нормально, но хотелось бы сохранить Mandriva.

Не знаю, почему Mandriva так бомбит мой жесткий диск, но хочу это пресечь. С радостью приму любые советы.

Дэйв Притчард [Dave Pritchard]

О Удаление *Kat* – первое, что приходит в голову. *Kat* индексирует все файлы в Вашей домашней директории, а это значит, что самый первый его запуск опустит Ваш быстрый 64-разрядный компьютер до старичка Sinclair Spectrum. Однако есть и другие программы периодического сканирования жесткого диска, например, *UpdateDB*, которая строит базу данных для *locate*. Программа запускается через *Cron* – по умолчанию, в Mandriva её запуск запланирован раз в неделю, рано утром, так что обычно активность диска проходит незамеченной. А если у Вас установлен пакет **anacron**, то он запускает все задачи *Cron*, не выполнявшиеся из-за того, что компьютер стоял выключенным – тут диск и может перевозбудиться.



Заметив повышенную активность диска, запустите **top** и посмотрите, кто отъедает больше всего процессорного времени. Если это *UpdateDB*, Вам не о чем беспокоиться: база данных обновляется раз в неделю, обычно всего за несколько минут. Попробуйте на какое-то время оставить компьютер включенным. Если индикатор не перестанет бешено мигать, скажем, через четверть часа – что-то неправильно, нужно продолжить поиски виноватого через *top*.

Быть может, индикатор сообщает о деятельности шины IDE, а вовсе не жесткого диска: например, проверке наличия DVD-диска в приводе. Это может быть и безобидная утилита вроде *Partmon*, сообщающая об отсутствии свободного места на Ваших разделах. Её можно отключить через Центр Управления Mandriva (Система > Сервисы). ***НБ***

Ну, работай же!

В Linux я относительно недавно. Я установил свой первый дистрибутив с диска *LXF* и сильно захотел установить *VMWare Player* и попробовать другие дистрибутивы. Но инструкции из *LXF78* по его установке в моем случае не сработали. Я успешно распаковал файл **VMWare-player-1.0.1-19317.tar.gz**, всё по журналу. Зато следующий шаг (**./configure**) явно не удался. Я уверен, что нахожусь в нужной директории (**VMWare-player-distrib**), однако набрав **./configure**, получаю ошибку:

```
bash: ./configure: No such file or directory
```

Я думаю, что-то не в порядке с моей SUSE 9.1. Наверное, в системе нет команды *configure*.

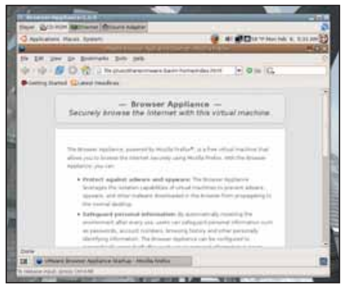
Джим [Jim]

О Скрипт *configure* включен во множество пакетов с исходными текстами – **./** означает выполнение команды из текущей директории, так что Вы запускаете команду прямо из распакованного архива, и система тут ни при чём. Большинство Linux-программ используют этот скрипт для проверки наличия в системе всех необходимых зависимостей. В Вашем случае он не сработал по причине того, что *VMWare Player* – это уже скомпилированная программа, с другим методом установки.

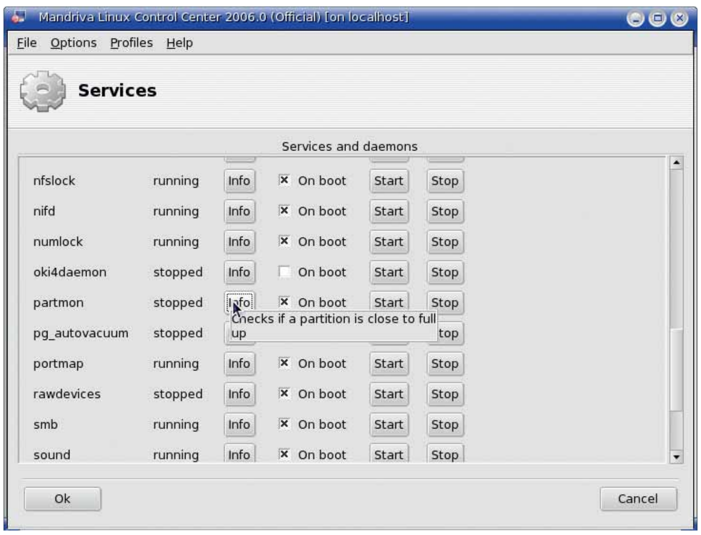
Статья про *VMWare* в том же журнале указывала, что нужно запустить **VMWare-install.pl**. Возьмите за правило просматривать раздел «Диск Linux Format» и директорию самого диска, всегда содержащую инструкцию по установке. Вот команды, необходимые для установки пакета после распаковки:

```
cd VMWare-Player-distrib
./VMWare-install.pl
```

Вам может понадобиться компилятор C и исходные тексты ядра, поскольку в ходе установки *VMWare* инсталлирует в систему готовый модуль ядра, а если готового модуля не находится, то сам собирает подходящий. Все это Вы можете установить через *Yast*. Нужные Вам пакеты называются gcc и **kernel-source**. ***НБ***



У двоичных пакетов и пакетов с исходными текстами разная установка.



Мыслите комплексно: мигание индикатора жесткого диска может быть вызвано утилитой *Partmon*, усердно проверяющей память разделов.

Linux запоздалый

В Привет вам из США и огромное спасибо за отличный журнал! Я прямо наслаждаюсь, читая все статьи и обзоры. Жаль только, что приходится ждать, пока журнал появится в местных магазинах...

У меня всего один вопрос к экспертам: хотелось бы услышать ваши рекомендации по дистрибутиву для ноутбука IBM T22 с 256 МБ памяти. Я попробовал Damn Small Linux, он отлично работает с моим Win-модемом. Можно ли его установить на жесткий диск?

Даррелл Ноблс [Darrell Nobles]

О Ваша первая проблема решается легко оформите подписку, и получите журнал всего через несколько дней после его появления в наших магазинах.

Ноутбуки от IBM отлично поддерживают Linux, поскольку IBM предоставляет все спецификации и драйвера. Большинство LiveCD можно установить на жесткий диск, однако DSL – не лучший выбор для Вас. DSL предназначен сугубо для использования как LiveCD, и его обновление может вызвать трудности.

Я бы посоветовал перебрать несколько LiveCD и взять тот, который будет работать с Вашим оборудованием лучше всего. Вот несколько примеров:

- **Knoppix** Установив его, Вы получите модифицированную версию Debian. **www.knoppix.com**
- **Ubuntu или Kubuntu** Оба основаны на Debian: первый использует графическое окружение Gnome, второй – KDE. **www.ubuntu.com**
- **PCLinuxOS** LiveCD с возможностью установки и простым обновлением. **www.pclinuxos.com**
- **Kanotix** Еще один LiveCD на базе Debian с невероятно простой процедурой установки. **http://kanotix.com**

НБ

>>

★ ВОПРОС МЕСЯЦА!

Победитель конкурса на лучший вопрос в августе **Анатолий Евдокимов** получает заслуженный приз – подарочный сертификат на 1000 рублей от интернет-магазина LinuxCenter.Ru

Просим Анатолия связаться с редакцией по адресу: **info@linuxformat.ru**

И снова о Microsoft

В Как известно, большинство современных дистрибутивов позволяют обновлять и устанавливать программы через сеть, но иногда это сложно сделать из прокси-сервера компании. У нас, например, используется Microsoft ISA с NTLM-авторизацией, и если в Windows достаточно установить клиент ISA, то в Linux это представляет проблему, поскольку стандартным способом соединение с прокси-сервером не устанавливается.

Подскажите, как же в Linux подключаются к прокси типа ISA с NTLM-авторизацией?

Анатолий Евдокимов

О Во-первых, NTLM-авторизация поддерживается браузерами семейства Mozilla: *Seamonkey* и *Firefox*, то есть при обычном серфинге с использованием этих приложений Вы вообще не должны испытывать никаких проблем.

Если, по какой-то причине, браузер не может пройти авторизацию, или Вам необходимо использовать другую программу, существует универсальный способ. Наш соотечественник Дмитрий Розманов разработал NTLM Authorization Proxy Server - программу на Python, которая перехватывает обращения к прокси-серверу и дополняет их необходимыми заголовками. Сайт программы расположен по адресу: **http://ntlmaps.sourceforge.net/**. Поскольку Python наверняка включен в Ваш дистрибутив, маловероятно, что Вы испытаете какие-либо проблемы с зависимостями. NTLMAPS работает как локальный прокси-сервер (по умолчанию используется порт 5865) и перенаправляет запросы ISA-серверу. Скачайте с сайта архив с программой распакуйте его, затем откройте файл server.cfg и внесите необходимые изменения: в поле **PARENT_PROXY** и **PARENT_PROXY_PORT** укажите адрес и порт ISA-сервера, укажите параметры авторизации. Сервер запускается командой **./main.py** (из каталога, в котором он распакован). После того, как сервер запущен, Вы можете открыть браузер/утилиту обновления дистрибутива, найти в ней настройки прокси-сервера и ввести туда 127.0.0.1:5865 (если Вы не меняли локальный прокси-порт в **server.cfg**).

Более подробное руководство по использованию NTLMAPS (на понятном английском) можно найти по адресу, приведенному в самом низу врезки.

ВС

http://www.linux.com/howtos/Web-Browsing-Behind-ISA-Server- HOWTO-4.shtml

Пауки в Сети

В У меня небольшой web-сайт, и похоже, что большинство его входящего трафика генерируется какими-то поисковиками. Как я могу защитить доступ к своему серверу *Apache* от потенциально вредоносных агентов, web-пауков и т.д.?

Р. Дэвидсон [R Davidson]

О Web-пауки могут быть использованы для кражи контента и информации о структуре вашего сайта, не предназначенной для посторонних глаз. Эти агенты также используются поисковыми движками для индексирования содержимого сайтов. Всё это неплохо, однако если Вы не хотите, чтобы Вас находили через поисковики, хорошо бы заблокировать агентам доступ. Большинство благонамеренных агентов не лезет дальше файла **robots.txt** в корневой директории сайта. Если они наглеют, нужны суровые меры.

В качестве первого шага можно сделать блокировку через HTTP-заголовок, это не непреодолимая преграда, но в большинстве случаев подойдет. Измените **webcopier** на строку, посылаемую пауком. Попробуйте так:

```
setenvif User-Agent ^webcopier
block
<Limit GET POST>
Order Allow,Deny
Allow from all
Deny from env=block
</Limit>
```

Или вот так:

```
RewriteEngine On
RewriteCond %{HTTP_USER_
AGENT} ^WebCopier [NC,OR]
RewriteRule ^.* - [F,L]
```

КК

Пропавшая библиотека

В При попытке сборки многих программ из исходных текстов команда **./configure** выдает следующее сообщение о ошибке :

checking for Qt... configure: error: Qt (>= Qt 3.2) (library qt-mt) not found.
Please check your installation!
For more details about this problem, look at the end of config.log.
Make sure that you have compiled Qt with thread support!

Исследования Интернета показали, что это проблема встречается у многих, но чёткого ответа я не нашёл. Ясно, что конфигуратор не может найти библиотеку **qt-mt**, хотя всевозможные пакеты ***.devel** установлены, да и сами библиотеки присутствуют. Использование параметра **--with-qt-libraries** проблемы не решает.

Я использую 64-битную версию OpenSUSE 10.1 на Athlon-64 3200+.

Может быть, вы поможете решить эту проблему?

Владимир Синотов

О После длительной переписки с автором вопроса и проверки мыслимых и немыслимых предположений, обнаружилось, что ключ к разгадке лежал практически на поверхности. Исследование файла **config.log** показало, что компоновщик пытался использовать 32-битные версии библиотек, что, естественно, было обречено на провал. Простой параметр **--enable-libsuffix=64**, переданный сценарию **configure**, полностью решает проблему.

Редакция *LXF* выражает благодарность Владимиру Синотову, который не просто ждал от нас готового ответа, но активно участвовал в процессе решения этой проблемы и, в конечном итоге, обнаружил «недостающий элемент мозаики». ***ВС*** LXF

СПРАШИВАЙТЕ ПРАВИЛЬНО!

- Пожалуйста, не забывайте сообщать все необходимые данные о вашей системе. «У меня не работает X» нам мало что скажет, если мы не знаем, какую версию X вы имеете в виду и на какой аппаратной конфигурации пытаетесь ее запустить.
- Опишите свою проблему с максимальной точностью. Причитания типа «Оно не работает» или «У меня ошибка» вряд ли дадут нам возможность помочь Вам. Каким именно образом нечто не работает? Чего вы от него хотели? Какое поступает сообщение об ошибке?
- Помните, пожалуйста, что люди, работающие в нашем журнале, НЕ являются авторами или разработчиками Linux или Вашего конкретного пакета (дистрибутива). Иногда нужная Вам информация имеется на соответствующем сайте. Ознакомьтесь с документацией!

Мы стараемся ответить на все вопросы. Если вы не нашли ответа на свой, просмотрите другие ответы – может быть, там разбирается проблема, аналогичная вашей. Все вопросы, к сожалению, поместить невозможно.

ОТДЕЛЬНО – ОБ УСКОРЕНИИ САЙТОВ

Оптимальный SQL

В У меня есть web-сайт с динамическим содержимым, генерируемым из базы данных MySQL. Статические страницы открываются очень быстро, а вот динамические ужасно тормозят. Как я могу увеличить скорость работы сайта?

Х. Брук [H Brooke]

О Разработчики MySQL включили несколько готовых образчиков файла **my.cnf**, оптимизированных под различное количество используемой оперативной памяти. Для их поиска выполните

```
locate cnf lgrep my-
```

Использование одной из этих конфигураций может помочь увеличить скорость.

Но более важной является оптимизация Вашего SQL-кода. Для определения виновников замедления Вы можете создать отдельный файл журнала:

```
touch /var/log/mysql-slow.log
chown mysql:mysql /var/log/mysql-slow.log
```

Обратите внимание, что им должен владеть пользователь, запускающий mysqld. Теперь добавим в секцию **[mysqld]** файла **/etc/my.cnf** следующую строку для слежения за медленными запросами (а потом перезапустим MySQL):

```
log-slow-queries=/var/log/mysql-slow.log
```

Вот пример записи в журнале:

```
# Time: 030207 15:03:33
# User@Host: wsuser[wsuser] @ localhost.localdomain [127.0.0.1]
# Query_time: 13 Lock_time: 0 Rows_sent: 117 Rows_examined: 234
use wsdb;
SELECT l FROM un WHERE ip='209.xx.xxx.xx';
```

Здесь видно, что wsuser обращался к базе wsdb через localhost (т.е. локально), и база рассмотрела 234 строки, это заняло 13 секунд.

Попробуйте оптимизировать наиболее частые и медленные запросы. Проверьте значение Rows_examined для каждого медленного запроса. Если это число хотя бы в два раза больше суммы всех строк в каждой опрашиваемой таблице, нужно добавить индексирование. Правда, при этом возникают лишние операции сравнения, отрицательно влияющие на производительность. КК

Объяснить и описать

В Динамические данные на моем сайте обрабатываются слишком медленно. Я подключил слежение за медленными запросами. Расскажите, что делать дальше и как их исправить.

Р. Трент [R Trent]

О Главное – определить причину медленности запросов и оптимизировать их. Если в Вашем журнале медленных запросов слишком много, скачайте mysql_slow_log_parser (**http://tinyurl.com/o2qek**), чтобы выявить, какие из них повторяются чаще всего.

Определив такие запросы, используйте *explain* (объяснить) и *desc* (*describe*, описать) для определения любой возможной оптимизации. Пусть у меня есть две одинаковые таблицы по два столбца, без индексов или ключей (см. вывод в **Таблице 1**):

```
mysql> desc t1;
```

Обе содержат по 10000 строк данных. У Вас в журнале может повторяться запрос:

```
# Query_time: 29 Lock_time: 0 Rows_sent: 1 Rows_examined: 10000
use lxf;
select count(*) from t1, t2 where t1.jk= t2.jk;
```

Используя запрос *explain*, давайте посмотрим, как работает SQL (вывод в **Таблице 2**).

```
mysql> explain select * from t1, t2 where t1.jk = t2.jk;
```

Это значит, что проверено 7,191,637 строк в таблице t2. Потом эти же 7,191,637 строк проверяются в таблице t1. Далее, согласно *describe* (desc) по таблице (и полям NULL), можно заметить, что в таблицах нет ключей и индексов. Давайте добавим простой индекс в обе таблицы:

```
mysql> create index t1jk on t1(jk);
mysql> create index t2jk on t2(jk);
```

Теперь запрос *describe* покажет индекс (вывод в **Таблице 3**). Нам нужно опять посмотреть, сколько строк будет проверено в следующем запросе (вывод в **Таблице 4**).

```
mysql> explain select count(*) from t1, t2 where t1.jk= t2.jk;
```

Проверка таблицы t2 с использованием индекса t2j1 свелась к проверке 10000 строк. Теперь сравним их с 50 строками в таблице t1, возвращённых сортировкой по индексу t1j1.

Мы можем очистить кэш и перезапустить запрос:

```
mysql> flush query cache;
mysql> select count(*) from t1, t2 where t1.jk= t2.jk;
mysql> select count(*) from t1, t2 where t1.jk= t2.jk;
+----------+
 count(*) l
+----------+
 498167 l
+----------+
1 row in set (0.66 sec)
```

Ранее занимавший 29 секунд, теперь он выполняется меньше чем за секунду. Этот несложный пример даёт метод, который может быть использован при оптимизации любых повторяющихся запросов без ключей и индексов для важных колонок. КК



Таблица 1)

Field	Type	Null	Key	Default	Extra
jk	varchar(20)	YES		NULL	
largeval	varchar(30)	YES		NULL	

Таблица 2)

id	select_type	table	type	possible_keys	key	key_len	ref	rows	Extra
1	SIMPLE	t2	ALL	NULL	NULL	NULL	NULL	7191637	
1	SIMPLE	t1	ALL	NULL	NULL	NULL	NULL	7191652	Using where

Таблица 3)

Field	Type	Null	Key	Default	Extra
jk	varchar(20)	YES	MUL	NULL	
largeval	varchar(30)	YES		NULL	

Таблица 4)

id	select_type	table	type	possible_keys	key	key_len	ref	rows	Extra
1	SIMPLE	t2	index	t2jk	t2jk	21	NULL	10000	Using index
1	SIMPLE	t1	ref	t1jk	t1jk	21	lxf.t2.jk	50	Using where; Using index

Диск Linux Format

Долгожданный релиз SUSE 10.1 теперь можно установить с нашего DVD!

ВСЁ ОТНОСИТЕЛЬНО



Майк Сондерс разнёс Пола за бильярдом. Потом Пол победил в Tetris. Так и надо!

Я везучий: 100% моей работы можно выполнить, используя открытое ПО. DVD журнала *LXF* создаётся под Linux, наш web-сайт работает под Linux, а эта колонка была набрана в *Nano*. Дома я развлекаюсь с другими открытыми ОС, например, Syllable и Haiku – короче, никакие мои занятия не требуют закрытых программ. Всё в порядке, мир во всём мире.

Однако, по-моему, важно быть в курсе дел и в других системах, в частности, в Windows. Малость походив по Slashdot, вы обнаружите

«ПРОБЛЕМЫ LINUX ВЫГЛЯДЯТ ТАКОЙ МЕЛОЧЬЮ НА ФОНЕ WINDOWS.»

дебаты Linux-против-Windows, где линуксоиды восклицают, что у детища Microsoft «экран смерти» возникает каждые 15 секунд. Но дни Windows ME давно прошли! XP, возможно, не идеальная система, но она гораздо надёжнее, чем уверяют наши спорщики. Обычные пользователи Windows, увидав подобные заявления, могут посчитать коллег из Linux просто упёртыми фанатиками.

Чтобы продвигать Linux, мы должны быть хорошо информированы; кроме всего прочего, это помогает видеть перспективу. Как почти 100% пользователь Linux в течение нескольких лет, я могу брюзжать о некоторых аспектах этой системы (например, об установке программ), но каждый раз, когда я вынужден обращаться к Windows, на этом фоне проблемы Linux начинают выглядеть тривиальными. Когда вы в следующий раз поссоритесь с Linux, посетите новичка, пользующегося компьютером Windows, нашпигованным шпионскими программами, и сразу почувствуете себя хорошо.

mike.saunders@futurenet.co.uk

ДИСТРИБУТИВЫ

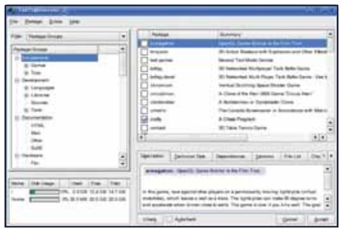
SUSE 10.1 OSS

SUSE постоянно входит в число самых популярных дистрибутивов – точнее, по рангу популярности Distrowatch, он идет сразу после лидера Ubuntu (по данным за последние 12 месяцев). Хотя это один из самых старых дистрибутивов – его первый релиз вышел ещё в 1994-м году – он по-прежнему завоёвывает себе поклонников, и вполне заслуженно. Опытная команда инженеров, корпоративное качество Novell и активное сообщество делают SUSE одним из лучших доступных дистрибутивов.

Это второй релиз, производный от OpenSUSE (**www.opensuse.org**) – открытого проекта, который спонсировала Novell, чтобы улучшить взаимодействие с пользователями и разработчиками SUSE. SUSE – более чем способная серверная система, но истинное её призвание – настольные и рабочие станции. В версии 10.1 была проделана серьёзная работа, чтобы рабочий стол выглядел максимально гладким и интегрированным, независимо от того, используете ли вы KDE или Gnome (выбрать между ними можно на этапе установки, или после установки добавить любой рабочий стол с помощью *Yast*). Вы обнаружите одну и ту же тему повсюду – взгляните, например, на заставку *Gimp* – а если хватит смелости, добавьте *Xgl*, у него в изобилии радующих глаз мулек (*Xgl* всё ещё считается нестабильным, поэтому по умолчанию он не включен).

Крутые инновации

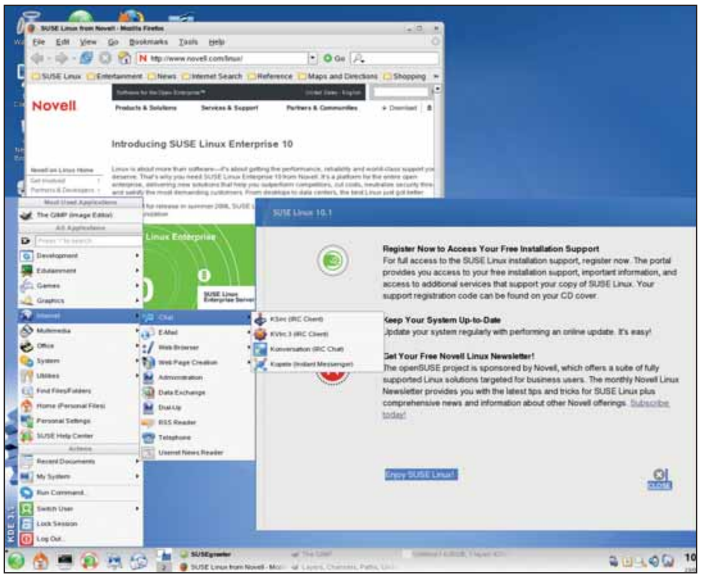
Из ярких особенностей этого дистрибутива нужно отметить *NetworkManager*, который делает подсоединение к сети и перемещение между сетями таким простым, что проще не бывает, а также *AppArmor* – систему безопасности Novell, разработанную с целью ограничения прав приложений на вашем компьютере, чтобы уменьшить последствия атаки. В *Yast* интегрирована конфигурация *Xen 3*, а все основные пакеты обновлены до новых, стабильных релизов, включая KDE 3.5.1, Gnome 2.12.2 и *OpenOffice.org 2.0.2*. Пока отзывы о новом релизе весьма позитивны, так что если вы используете версию 10.0, то стоит обновиться.



После установки, можете добавить себе программ с нашего DVD, запустив *Yast* и выбрав пункт Software Management.

На нашем DVD вы найдёте полную версию SUSE 10.1, готовую к установке – просто загрузите ПК с этого диска! Через страницу вы найдёте пошаговое руководство по установке. Имейте в виду, что для установки SUSE ваш компьютер должен обладать минимум 256 МБ памяти, процессором типа Pentium или новее и 3 ГБ дискового пространства для стандартного набора пакетов.

Опции загрузки по умолчанию должны устроить большинство пользователей. Если вы встретитесь с проблемой загрузки с DVD, попробуйте выбрать режим Safe Settings в главном меню. Если вы подозреваете, что проблема в вашем ПК, можете выполнить тест памяти или нажать кнопку F3 и изменить разрешение экрана инсталлятора (пригодится, если во время старта возникли проблемы в видеокартой).



SUSE 10.1 и KDE. Три клиента IRC – есть из чего выбрать!

НЕТ DVD-ПРИВОДА?

Если вам нужно установить SUSE на компьютер, у которого есть только CD-дисковод, вам повезло. Мы включили в наш DVD-диск систему *Jigdo*, благодаря которой можно сделать ISO-образы для CD-диска, записать их на CD-R, загрузить ваш ПК с первого диска и начать обычный процесс установки. (Yast будет предупреждать вас, что нужно вставить очередной диск). Об использовании *Jigdo* вы можете узнать подробнее, почитав файл **index.html** – там под введением приводятся объяснения.

Проблемы с правами доступа при запуске *Jigdo* означают, что ваш DVD примонтирован с опцией 'noexec'. Быстро исправить ситуацию можно, скопировав файл **Essentials/Jigdo/jigdo-file** с диска в каталог, упомянутый в PATH, например, **/usr/bin**, и попробовав ещё раз. Если что, добро пожаловать в наш форум.

СОДЕРЖИМОЕ ДИСКА

Журнал

3D-Games	Файлы из обзора 3D-игр.
OOo BASIC	Дополнительный код из учебника.
Roundup	Эмуляторы терминалов.

Рабочий стол

Beagle 0.2.6	Поисковая система для рабочего стола.
Compiz CVS	Оконный менеджер для Xgl.
EPDFView 0.1.3	Просмотрщик PDF-файлов.
SuperKaramba 0.39	Интересные виджеты для рабочего стола.
Xgl CVS	X-сервер с 3D-эффектами.
Xpdf 3.01	Просмотрщик PDF-файлов.

Разработка

KDESVN 0.9.0	Клиент Subversion для KDE.
PHP 5.1.4	Язык программирования.
PHP-Tutorials	Учебники от LXF.
Subversion 1.3.1	Система контроля версий.
Vim 7.0	Текстовый редактор.

Дистрибутивы

SUSE 10.1	Дистрибутив Linux от Novell.

Игры

Empty Clip 1.0.1	Двумерная ролевая игра.
Level Shmup 1.0	Стрелялка.
ManiaDrive 1.01	Трюковые гонки.

Жемчужинки

Deskbar 2.14.2	Утилита поиска для Gnome.
EasyTag 1.99.12	Расстановка тэгов для музыкальных файлов.
Incollector 0.1	Менеджер коллекций.
Ion 2/3	Оконный менеджер.
Links 1.00p16	Текстовый web-браузер.
Mp3togo 0.5.1	Изменение размеров MP3-файлов.
RSS-GLX 0.8.1	Коллекция хранителей экранов.
Skippy 0.5.0	Переключатель задач в стиле Exposé.
Tea 13.3	Текстовый редактор.
Tomboy 0.3.5	Ведение записей в стиле Wiki.

Графика

DevIL 1.5.6	Межплатформная библиотека для работы с изображениями.
GEGL CVS	Библиотека обработки изображений.
Gimp 2.2.11/2.3.8	Графический редактор.
Inkscape 0.44	Векторный графический редактор.
Ogre 1.2.0	Графический движок.

HotPicks

AckerTodo 3.6	Менеджер списка дел.
Avidemux 2.1.2	Утилита редактирования видео.
Bonfire 0.3.1	Приложение для создания CD/DVD.
Byzanz 0.1.1	Запись с экрана.
Goupil 0.1.0	Менеджер членства в клубе.
Medit 0.6.98	Текстовый редактор.
NoFriction 0.1	Игра-головоломка.
Pipepanic 0.1.3	Игра-головоломка.
SVGpage 0.4	Конвертер изображений.
Visopsys 0.62	Операционная система.

Интернет

Mozilla Firefox 2 Beta 1	Бета-версия Firefox 2.
Gossip 0.11	Чат-клиент.

Офис

Gnumeric 1.7.0	Электронная таблица.

Безопасность

FwBuilder 2.0.12	Утилита конфигурирования брандмауэра.
Nmap 4.03	Сетевой сканер.
Sussen 0.22	Сканер безопасности.

Сервер

Amanda 2.5.0p2	Система резервного копирования.
Asterisk 1.2.8	Система VACT с открытым кодом.
KPoGre 1.3.8	Интерфейс к базе данных.

Звук

AmaroK 1.4.1	Аудиопроигрыватель KDE.

Система

KleanSweep 0.2.8	Очистка дискового пространства.
Syllable 0.6.1	Уникальная настольная система.
Video-drivers	Драйвера ATI и Nvidia.

Essentials

Avifile 0.7.43	Библиотека чтения и записи AVI файлов.
CheckInstall 1.6.0	Создание бинарных пакетов.
Coreutils 5.96	Утилиты командной строки.
CSV	Индексные файлы Coverdisc.
GLib 2.8.6	Низкоуровневая библиотека.
GTK 2.8.18	Инструментарий для интерфейса пользователя.
Jigdo	Конструктор ISO-образов.
Kernel 2.6.17.6	Последний релиз ядра Linux.
libsigc 2.0.17	Система обратных вызовов на C++.
libXML 2.6.24	XML-анализатор и инструментарий.
Ncurses 5.5	Инструментарий для создания окон в текстовом режиме.
Python 2.4.3	Язык программирования.
Rawrite	Запись образов на дискеты.
SBM 3.7	Умный загрузчик (The Smart Boot Manager).
SDL 1.2.10	Мультимедиа-библиотека.



***ОС Syllable* – сплошной дизайн, интеграция и простота использования.**

РАЗРАБОТКА

УЧЕБНИКИ ПО PHP

В прошлом месяце мы свели из нашего журнала подборку учебников по *Gimp* и записали её на диск. Теперь очередь за PHP – мы предоставляем вам полное собрание учебников (от *LXF30* и до *LXF82*!). Учебники записаны в формате PDF, поэтому их можно прочесть почти в любом дистрибутиве; если у вас нет читалки PDF, попробуйте *Xpdf*, из раздела Рабочий стол.

PHP – один из самых популярных скриптовых языков для web: на нём можно написать сложный модульный сайт (пример – сайты на базе *PostNuke*), его легко изучить и у него очень активное сообщество пользователей. Синтаксически PHP похож на C и Java, и если вы когда-то программировали, то быстро его изучите. Если вы не кодировали сроду, то возрадуйтесь – PHP является одним из лучших языков для начинающих. На нашем DVD вы найдёте полный набор учебников из раздела **Разработка**, который ведёт Пол Хадсон (автор книги *PHP in A Nutshell*). Скорее всего PHP имеется в репозитарии вашего дистрибутива, но если вы не сможете его найти, в разделе **Разработка** нашего диска вы найдёте исходный код последнего релиза.

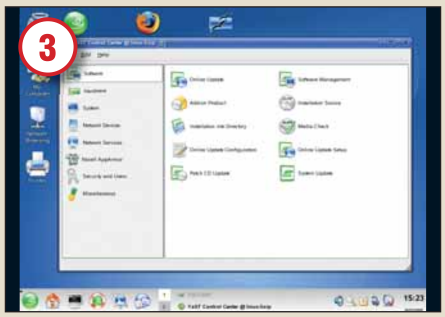
ИССЛЕДУЙТЕ НОВЫЙ РАБОЧИЙ СТОЛ SUSE 10.1



Во время установки вас спросят, что вы хотите установить в качестве рабочего стола: KDE или Gnome. Мы рекомендуем KDE, он лучше сочетается с остальной частью дистрибутива. После загрузки вы увидите этот рабочий стол – кликните на зелёную кнопку в нижнем левом углу для вызова главного меню. На нижней панели вы найдёте панель задач (со списком стартовавших программ) и переключатель, для навигации между виртуальными рабочими столами.



Чтобы сконфигурировать рабочий стол, используйте KDE Control Centre (кликните Menu > Personal Settings). Здесь можно менять цветовую схему, шрифты, декорации окон и стиль виджетов. Собираетесь имитировать вид CDE или Windows 9x? И это тоже можно, можно также включить всякие эффекты вроде прозрачности и теней. Имейте в виду, что сама инсталляция SUSE настраивается не здесь, для этого вам нужно…

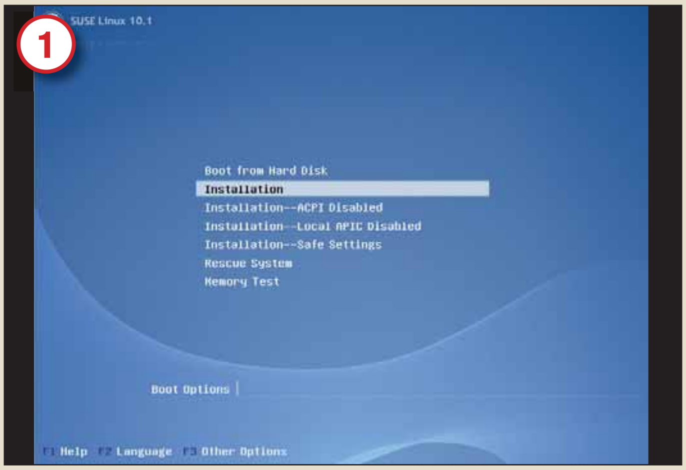


Запустить *Yast* – многоцелевой инструмент и одну из замечательных особенностей SUSE Linux. Для вызова этого приложения кликните на Menu > System > Yast, после чего выберите нужную опцию. С помощью *Yast* можно установить новые программы, сконфигурировать аппаратное обеспечение, поменять настройки загрузчика и много чего ещё. *Yast* также отлично работает в других рабочих столах и оконных менеджерах, так что если у вас *Fluxbox*, но вы хотите контролировать систему, можете и тут воспользоваться *Yast*.

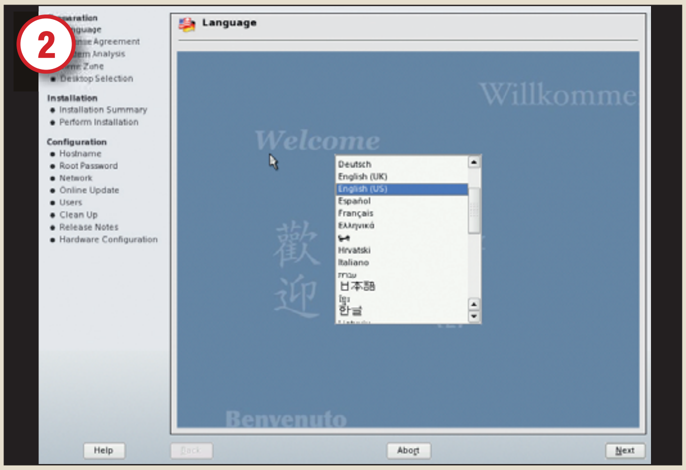
»

УСТАНОВКА SUSE LINUX 10.1

Установка SUSE 10.1 почти всегда проходит гладко, но если вы столкнулись с трудностями, посетите www.suseforums.net или www.linuxforum.ru.



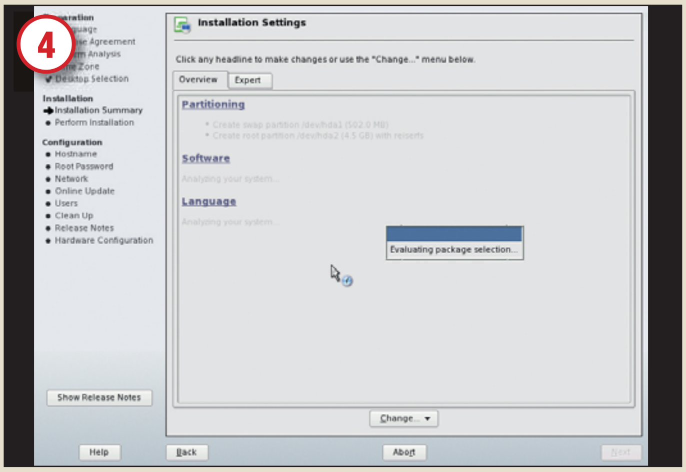
Для установки SUSE загрузите ваш компьютер с DVD (возможно, понадобится поменять порядок загрузки в BIOS). Когда появится это меню, нажмите один раз кнопку курсора вниз и затем Enter, чтобы выбрать пункт Installation.



По окончании сообщений ядра появится графический инсталлятор, и вы окажетесь в этом экране. Выберите свой язык, нажмите Next в нижнем правом углу экрана и прочтите лицензионное соглашение.



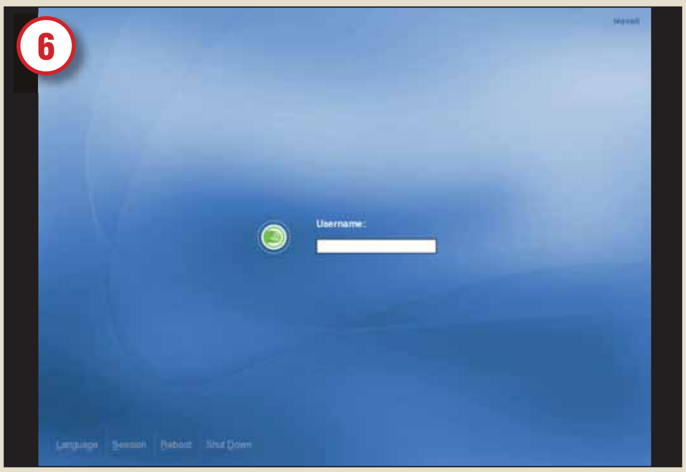
SUSE проанализирует вашу систему и предложит вам установку с нуля или обновление, если у вас уже была предыдущая версия SUSE. Укажите ваш часовой пояс и выберите рабочий стол (мы рекомендуем KDE).



Инсталлятор запустит процесс настройки разделов жёстких дисков. Если на вашем жёстком диске нет других ОС, просто щёлкните Accept; в противном случае щёлкните Partitioning и выделите раздел под Linux (в идеале не меньше 3 ГБ).



После копирования файлов SUSE перезагрузится, и начнётся загрузка с жёсткого диска и финальные действия инсталлятора, который определит оборудование, позволив вам внести необходимые изменения, и спросит у вас имя пользователя и пароль.



Процесс завершён! Теперь вы можете войти в систему с теми самыми именем и паролем (либо система впустит вас автоматически, если во время установки была выбрана соответствующая опция). Наслаждайтесь вашим SUSE 10.1!

Информация о диске

Внимательно прочтите это перед тем, как использовать DVD-диск.

ЧТО-ТО ПОТЕРЯЛИ?

Часто случается, что новые программы зависят от других программных продуктов, которые могут не входить в текущую версию вашего дистрибутива Linux.

Мы стараемся предоставить вам как можно больше важных вспомогательных файлов. В большинстве случаев, последние версии библиотек и другие пакеты мы включаем в каталог «Essentials» (Главное) на прилагаемом диске. Поэтому, если в вашей системе возникли проблемы с зависимостями, следует заглянуть именно туда.

ФОРМАТЫ ПАКЕТОВ

Мы стараемся включать как можно больше различных типов установочных пакетов: RPM, Deb или любые другие. Просим вас принять во внимание, что мы ограничены свободным пространством и доступными бинарными выпусками программ. По возможности, мы будем включать исходные тексты для любого пакета, чтобы вы смогли собрать его самостоятельно.

ДОКУМЕНТАЦИЯ

На диске вы сможете найти всю необходимую информацию о том, как устанавливать и использовать некоторые программы. Пожалуйста, не забывайте, что большинство программ поставляются вместе со своей документацией, поэтому дополнительные материалы и файлы находятся в соответствующих директориях.

ЧТО ЭТО ЗА ФАЙЛЫ?

Если вы новичок в Linux, вас может смутить изобилие различных файлов и расширений. Так как мы стараемся собрать как можно больше вариантов пакетов для обеспечения совместимости, в одном каталоге часто находятся два или три файла для различных версий Linux, различных архитектур, исходные тексты и откомпилированные пакеты. Чтобы определить, какой именно файл вам нужен, необходимо обратить внимание на его имя или расширение:

имя_программы-1.0.1.i386.rpm – вероятно, это бинарный пакет RPM, предназначенный для работы на системах x86;
имя_программы-1.0.1.i386.deb – такой же пакет, но уже для Debian;
имя_программы-1.0.1.tar.gz – обычно это исходный код;
имя_программы-1.0.1.tgz – тот же файл, что и выше по списку: «tgz» - это сокращение от «tar.gz»;
имя_программы-1.0.1.tar.bz2 – тот же файл, но сжатый bzip2 вместо обычного gzip;
имя_программы-1.0.1.src.rpm – также исходный код, но поставляемый как RPM-пакет для упрощения процесса установки;
имя_программы-1.0.1.i386.FC4.RPM – бинарный пакет RPM для x86, предназначенный специально для операционной системы Fedora Core 4;
имя_программы-1.0.1.ppc.Suse9.rpm – бинарный пакет RPM, предназначенный специально для операционной системы SUSE 9.x PPC;
имя_программы-devel-1.0.1.i386.rpm – версия для разработчиков.

Если диск не читается...

Это маловероятно, но если все же прилагаемый к журналу диск поврежден, пожалуйста, свяжитесь с нашей службой поддержки по электронной почте: disks@linuxformat.ru



Содержание LINUX FORMAT DVD Август 2006

Сторона 1

Рабочий стол
Beagle – настольная поисковая система
Compiz CVS – оконный менеджер для Xgl
EPDFView – просмотрщик PDF на GTK
SuperKaramba – утилита для отображения виджетов
XGL CVS – X-сервер с потрясающими эффектами
Xpdf – просмотрщик PDF

Разработка
KDESvn – Subversion-клиент для KDE
PHP – интерпретируемый язык
PHP-Tutorials – учебники по PHP
Subversion – система контроля версий
Vim – мощный текстовый редактор

Дистрибутивы
SUSE 10.1 – новая версия дистрибутива от Novell

Игры
EmptyClip – двумерная стрелялка
Level Shmup – аркадная стрелялка
ManiaDrive – автомобильный симулятор

Изумруды
Deskbar – поисковый инструмент для Gnome
EasyTAG – программа для работы с тегами
Incollector – менеджер коллекций
Ion – оконный менеджер.
Links – консольный веб-браузер.
Mp3togo – улитита для перекодировки MP3.
RSS-GLX – коллекция скринсейверов.
Skippy – менеджер задач вроде Expose.
TEA – текстовый редактор.
Tomboy – программа для создания заметок.

Графика
DevIL – кроссплатформенная библиотека для обработки изображений
GECL CVS – библиотека обработки изображений
Gimp – популярный графический редактор
Ogre – трехмерный движок

Горячие новинки
ackerTodo – менеджер Todo-листов.
Avidemux – утилита для редактирования видео.
Bonfire – программа для мастеринга CD/DVD.
Byzanz – утилита для записи видео с экрана.
Medit – текстовый редактор.
NoFriction – паззл.
Pipepanic – и еще один паззл.
SVGpage – конвертер изображений.
Visopsys – операционная система.

Интернет
Bon-Echo – предварительная версия Firefox 2
Gossip – Jabber-клиент для Gnome

Офис
Gnumeric – электронная таблица

Безопасность
FwBuilder – утилита для настройки файерволла
Nmap – сканер портов
Sussen – сканер уязвимостей

Поставляется вместе с журналом LINUXFORMAT номер 8(82) Август 2006

Сторона 2

Битрикс: Управление сайтом
Kororaa XGL LiveCD 0.2
OpenSource Format

Сервер
Amanda – система резервного копирования
Asterisk – открытый PBX-комплекс
KpoGre – интерфейс к базам данных

Звук
AmaroK – аудиоплеер

Система
KleanSweep – утилита для очистки диска
Syllable – операционная система
Video-Drivers – проприетарные драйверы для видеокарт NVIDIA и ATI

Комментарии? Присылайте ваши мысли и предложения по электронной почте: info@linuxformat.ru
Пожалуйста, ознакомьтесь с опубликованной в журнале инструкцией перед использованием данного диска.
Настоящий диск тщательно тестировался и проверялся на всех стадиях производства, однако, как и в случае с любым новым ПО, мы рекомендуем вам использовать антивирусный сканер. Мы также рекомендуем всегда иметь под рукой актуальную резервную копию данных вашего жесткого диска. К сожалению, редакция Linux Format не может принимать на себя ответственность за любые повреждения, разрушения или иные убытки, которые может повлечь за собой использование этого DVD, представленных на нем программ или данных. Перед тем, как устанавливать какое-либо ПО на компьютер, подключенный к сети, проконсультируйтесь с сетевым администратором.

Дефектные диски. В маловероятном случае обнаружения дефектов на данном диске, пожалуйста, обращайтесь по адресу: disks@linuxformat.ru

Тираж изготовлен ООО «Кардмедиа», Россия, Санкт-Петербург, 196006 ул. Цветочная д. 7, тел. +7 (812) 388-8290. Лицензия МПТР России ВАФ № 77-68

СОЗДАНИЕ УСТАНОВОЧНЫХ ДИСКОВ ПРИ ПОМОЩИ CDRECORD

Самый быстрый способ записать ISO-образ на чистую матрицу – это *cdrecord*. Для всех перечисленных ниже действий потребуются права root. Для начала определите путь к вашему устройству для записи дисков. Наберите следующую команду:

```
cdrecord -scanbus
```

После этого на экране терминала должен отобразиться список устройств, подключенных к вашей системе. SCSI-адрес каждого устройства представляет собой три числа в левой колонке, например, 0,3,0. Теперь вы можете с легкостью записать образ на диск:

```
cdrecord dev=0,3,0 -v /путь к образу/image.iso
```

Чтобы упростить дальнейшее использование cdrecord, сохраните некоторые настройки в файле **/etc/default/cdrecord**. Добавьте по одной строке для каждого устройства записи (вероятно, в вашей системе присутствует всего одно такое устройство):

```
Plextor= 0,3,0 12 16M
```

Первое слово в этой строке – это метка, затем, после адреса SCSI-устройства вы должны указать скорость и размер буфера. Теперь вы можете заменить SCSI-адрес в командной строке на выбранную вами метку. Все будет еще проще, если вы добавите следующее:

```
CDR_DEVICE=Plextor
```

Все, что вам теперь нужно для записи ISO-образа – это набрать команду

```
cdrecord -v /path/to/image.iso
```

Если вы не из числа любителей командной строки, в таком случае вам придет на помощь утилита *gcombust*. Запустите ее из-под root, выберите вкладку «Burn» и ISO 9660 Image в верхней части окна. Введите путь к образу, который вы хотите записать на диск, и смело нажимайте на «Combust!». Пока ваш образ пишется на диск, можете выпить чашечку кофе.

Другая ОС?

Вам не обязательно использовать Linux для записи компакт-диска. Все необходимые файлы уже включены в ISO-образ. Программы вроде *cdrecrod* просто переносят данные на чистую матрицу. Если у вас нет устройства для записи дисков, можно найти того, у кого оно есть, и записать диск на его компьютере. На нем может стоять Windows, Mac OS X, AmigaOS, или любая другая ОС.

Нет устройства для записи дисков?

А что если у вас нет устройства, с помощью которого можно было записать образ на диск? Вы знаете кого-либо с таким устройством? Вам не придется использовать Linux для записи дисков, подойдет любая операционная система, способная распознать привод записи дисков (см. выше).

Некоторые дистрибутивы умеют монтировать образы дисков и выполнять сетевую установку или даже установку с раздела жесткого диска. Конкретные методы, конечно, зависят от дистрибутива. За дополнительной информацией обращайтесь на web-сайт его разработчика. LXF

ДРУГИЕ ПРОГРАММЫ

НОВЫЕ РЕЛИЗЫ

Вышла версия *Amarok 1.4* – новый стабильный релиз очень популярного музыкального проигрывателя для KDE. Он любим настолько, что многие пользователи запускают его и под другими рабочими столами и оконными менеджерами. Пусть ему и нужны библиотеки KDE, но под Gnome на быстрой машине вы этого и не заметите. *Amarok 1.4* принёс массу дополнительных возможностей, включая улучшенную поддержку тэгов (так что ваши файлы WMA, MP4/AAC и RealMedia получат нормальные описания) и поддержку текстов песен (тексты можно скачать почти с каждого сайта).

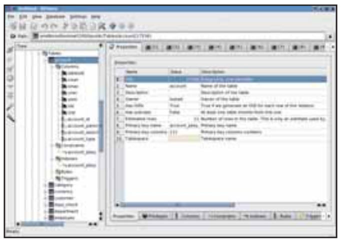
Другие заметные возможности – диалог статистики, для отображения ваших любимых песен и артистов, плюс улучшенная поддержка звуковых устройств, так что эта версия должна работать с обычными MP3-проигрывателями с интерфейсом USB. Данный релиз включил столько новых возможностей, что пропустить его просто нельзя; к счастью, в большинстве популярных дистрибутивов уже доступны двоичные пакеты.

На нашем DVD вы найдёте полный исходный код, вместе с RPM-файлами для Mandriva и SUSE. Если вы решите компилировать программу из исходных кодов, убедитесь, что у вас установлены пакеты разработчика KDE (обычно они называются **kdelibs-devel**).

Если вам нравится жизнь на переднем крае технологий, попробуйте *Gnumeric 1.7.0* – первый релиз электронной таблицы в ветке для разработчиков (см. раздел **Офис**). На этот релиз затрачен колоссальный труд, включая сотни обновлений и исправлений ошибок. Среди наиболее заметных новых свойств – импорт файлов *Microsoft Office* 12, линии регрессии в графиках и сильно улучшенный импорт документов OpenDocument. *Gnumeric 1.7.0* пока доступен только в вие исходного кода, его зависимости – библиотеки Gnome плюс *GOffice* (всё это есть на нашем DVD).

Все ваши базы данных

Не так давно обновился *KPoGre* – клиент для базы данных *PostgreSQL*. Мы чуть было не включили это обновление в обзор на стр. 56, но отказались от этой идеи, поскольку базы данных не самый распространённый тип приложений. *KPoGre* активно используется мастеров задач для администрирования – например, создания/модификации объектов базы данных (пользователей, таблиц и т.п.) и резервного копирования/восстановления баз данных. Исходный код и пакеты для SUSE записаны на наш DVD.

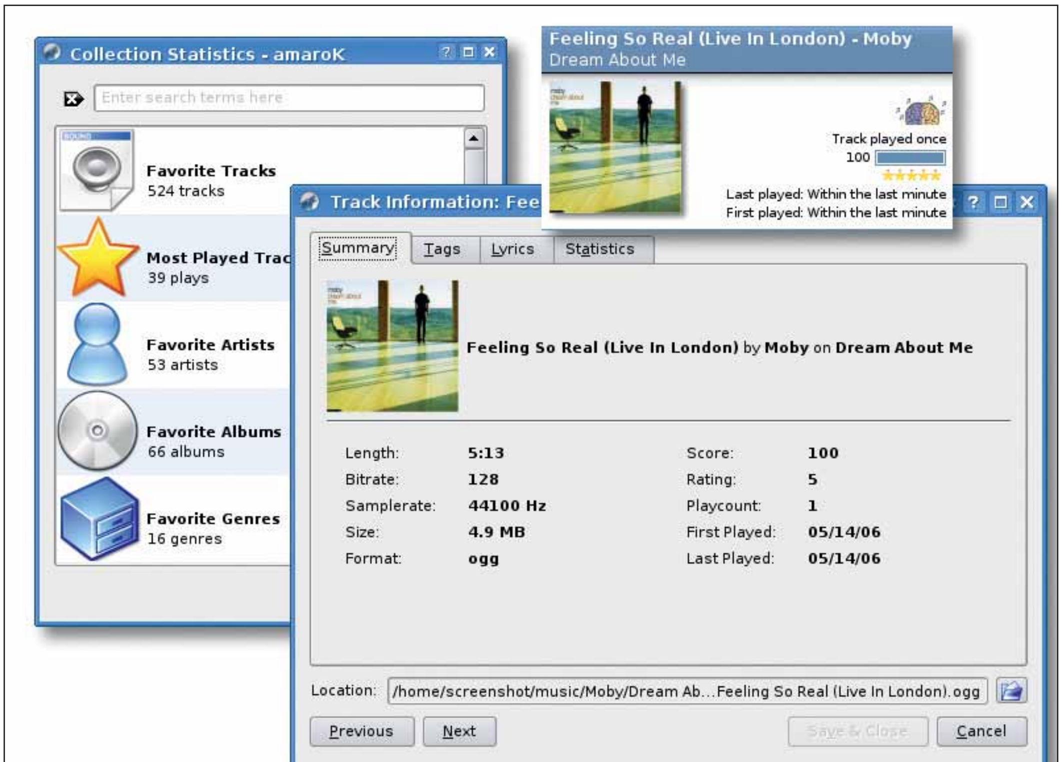


***KPoGre*: KDE-утилита управления *PostgreSQL*, с дурацким именем!**

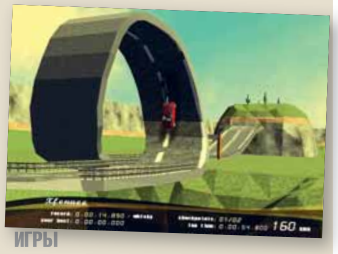
KleanSweep – удобная утилита для KDE, обзор которой был сделан в **HotPicks** *LXF74* – также обновилась. Эта утилита почистит вам жёсткий диск и удалит всякий мусор. Она ищет пустые файлы и каталоги, сломанные символические ссылки, позабытые исполняемые файлы и другой мусор, накапливающийся в часто используемой Linux системе. Сразу после установки свежей системы она вряд ли будет вам полезна, но если вы уже давно работаете с вашим дистрибутивом и добавляли-удаляли программы из репозитариев разных сортов, вам понравится порядок, который она наводит.

Конечно, на нашем DVD ещё много программ достойных внимания. Мы записали все приложения, рассмотренные в HotPicks, Тайных Кладах и в обзоре X-терминалов, а также утилиты безопасности, ПО для разработчиков и игры.

Ах да, если вы ищете полный индекс программ на этом DVD, обратитесь к каталогу *CSV* в разделе **Essentials**. Там находится коллекция файлов, которые вы можете импортировать в электронную таблицу и найти то, что вас интересует.



***Amarok* сработается с MP3-плейерами от Apple и iRiver, он поддерживает под-кастинг, собирает новую статистику, выставляет очки и работает с тэгами, и вдобавок имеет потрясающий Mood Bar (индикатор настроения)!**



ИГРЫ

Если (уж не знаем как) собираетесь пробовать это наяву, рекомендуем защитную дугу. И запаситесь валидолом.

MANIA DRIVE

Блиц-опрос: кто помнит фантастический *Stunt Car Racer* Джоффа Граммонда (Geoff Grammond)? Те, кто застал 8-битную эпоху, вероятно, играли в эту великолепную гоночную игру с невероятно опасными трассами и летающими развалинами. С тех пор мы навидались попыток воссоздать эту игру – например, серия *Destruction Derby* – но в них редко ощущался тот же сумасшедший драйв. Поэтому, когда мы наткнулись на *ManiaDrive*, мы не ожидали, что игра нас удивит – и ошиблись…

ManiaDrive – игра сумасшедшая. Постоянные вращения, прыжки, петли, переходящие в другие петли, и бесконечное издевательство над автомобилем. Вы управляете скромным красным автомобильчиком, ваша задача – проехать по безумному маршруту за минимальное время. 12 трасс нашпигованы всевозможными сложностями, затрудняющими контроль над автомобилем – съедете с магистрали не под тем углом, и вы уже подброшены в воздух и шансов на удачное приземление почти нет.

Чтобы запустить *ManiaDrive*, распакуйте файл **ManiaDrive-1.0.1-linux-i386.static-data.tar.gz**, перейдите в появившийся каталог и наберите **mania_drive.sh**. Кликните на Story, а затем на первую трассу, и начнется битва с секундомером (клавиши курсора контролируют автомобиль, а пробел вызывает рестарт). Подсказка: если лимит времени слишком жесткий и вы хотите попробовать другие трассы, отредактируйте их порядок в **gane/mania_drive.story**. LXF



LINUX FORMAT

Журнал зарегистрирован Федеральной службой по надзору за соблюдением законодательства в сфере массовых коммуникаций и охране культурного наследия
ПИ № ФС77-21973 от 14 сентября 2005 года
Выходит ежемесячно
Тираж 5000 экз.

РЕДАКЦИЯ РУССКОЯЗЫЧНОЙ ВЕРСИИ:

ГЛАВНЫЙ РЕДАКТОР
Валентин Синицын info@linuxformat.ru
Литературные редакторы
Елена Толстякова, Александр Толстой, Иван Мищенко
Переводчики
Александр Бикмеев, Павел Гладков, Светлана Кривошеина, Александр Кузьменков, Алексей Опарин, Сергей Салимов, Сергей Супрунов, Александр Толстой, Александр Черных, Юлия Шабунио, Павел Шер.
Допечатная подготовка
Мария Пучкова
Родион Водейко
Креативный директор
Станислав Медведев
Технический директор
Денис Филиппов
Директор по рекламе
Денис Игнатов +7 812 965 7236 advert@linuxformat.ru
Заместитель генерального директора
Софья Виниченко
Генеральный директор
Павел Фролов
УЧРЕДИТЕЛИ
частные лица
ИЗДАТЕЛИ
Павел Фролов, Станислав Медведев

Отпечатано в типографии «Текст»
ООО «ППК «Текст»
188680, Ленинградская область,
Всеволожский район, Колтуши, д.32
Заказ _____

Пре–пресс: *d.r.i.v.e-group*

РЕДАКЦИЯ АНГЛОЯЗЫЧНОЙ ВЕРСИИ:

Редактор Ник Вейтч (Nick Veitch) nick.veitch@futurenet.co.uk
Заместитель редактора Пол Хадсон (Paul Hudson) paul.hudson@futurenet.co.uk
Художественный редактор Эфрейн Херландез-Мендоза (Efrain Hernandez-Mendoza) efrain.hernandez-mendoza@futurenet.co.uk
Новостной редактор Майк Сондерс (Mike Saunders) mike.saunders@futurenet.co.uk
Литературный редактор
Ребекка Смелли (Rebecca Smalley) rebecca.smalley@futurenet.co.uk
Штатный автор
Грэм Моррисон (Graham Morrison) graham.morrison@futurenet.co.uk
Ассистент по выпуску
Эндрью Грегори (Andrew Gregory) andrew.gregory@futurenet.co.uk

АВТОРЫ
Джоно Бэкон (Jono Bacon), Марк Бейн (Mark Bain), Ладислав Боднар (Ladislav Bodnar), Нейл Ботвик (Neil Bothwick), Энди Ченнел (Andy Channelle), Дэвид Коулсон (David Coulson), Кингс Кобблер (Kings Cobbler), Майкл Дж. Хэммел (Michael J Hammel), Евгений Балдин, Андрей Боровский, Дмитрий Кирсанов, Петр Семилетов, Сергей Супрунов, Тихон Тарнавский, Алексей Федорчук, Сергей Яремчук.

ХУДОЖЕСТВЕННЫЙ ОТДЕЛ
Художники: Йон Блекшау (Jon Blackshaw), Эмит Петел (Amit Patel)
Фотографии: Corbis UK Ltd, Photodisc, Joby Sessions, SuperStock
Иллюстрации: Шейн Коллинж (Shane Collinge), Крис Винн (Chris Winn), Elly Walton Illustrations
Создание диска: Майк Сондерс (Mike Saunders)

КОНТАКТНАЯ ИНФОРМАЦИЯ
England: Linux Format, 30 Monmouth Street, Bath BA1 2BW
Tel 01225 442244 Email linuxformat@futurenet.co.uk
Россия: Санкт-Петербург, ул. Гончарная, 23, офис 54, телефон: (812) 717-00-37
Email: info@linuxformat.ru
Web: www.linuxformat.ru

В следующем месяце

LINUX FORMAT 8(82) АВГУСТ 2006



UBUNTU НАВСЕГДА!

Настольный Linux уже здесь: мы зовем его Ubuntu. Читайте эксклюзивное руководство о том, как выжать максимум из самого настольного дистрибутива современности!

Содержание номера может быть изменено



SpikeSource

Открытое ПО для бизнеса

Linux on Rails

Не пора ли переходить на Ruby?

ПОДПИСКА НА LINUX FORMAT

ПОДПИСКА В ЛИНУКСЦЕНТРЕ

Сколько стоит подписка?

Подписка на журнал «Linux Format» **12 номеров** (январь, февраль, март, апрель, май, июнь, июль, август, сентябрь, октябрь, ноябрь, декабрь) стоит 1800 рублей
Подписка на журнал «Linux Format» **6 номеров** (июль, август, сентябрь, октябрь, ноябрь, декабрь 2006 года) стоит 900 рублей

Как оформить подписку?

Чтобы оформить подписку на журнал «Linux Format», необходимо зарегистрироваться в интернет-магазине Linuxcenter.Ru, указав ФИО и подробный почтовый адрес подписчика, заказать товар «Подписка на журнал «Linux Format» 12 номеров 2006 года», или товар «Подписка на журнал «Linux Format» второе полугодие 2006 года», получить от системы квитанцию для оплаты в любом отделении Сбербанка (для физических лиц) или счет для оплаты по безналичному расчету (для юридических лиц)

Как оплатить подписку?

– по выставленному счету (для юридических лиц)
– по квитанции в любом отделении Сбербанка

Плюсы подписки

- подписка дешевле!
- гарантированное получение нового номера журнала!

ПОДПИСКА - 2006!

ПОДПИСКА ПО КАТАЛОГАМ РФ

Каталог агентства «РОСПЕЧАТЬ» – подписной индекс 20882

Каталог «ПРЕССА РОССИИ» – подписной индекс 87974

Ф. СП-1

Министерство связи РФ

АБОНЕМЕНТ на ~~газету~~ журнал

Linux Format

(наименование издания) (индекс издания)

Количество комплектов:

на 200_ год по месяцам

1	2	3	4	5	6	7	8	9	10	11	12

Куда (почтовый индекс) (адрес)

Кому (фамилия, инициалы)

ДОСТАВОЧНАЯ КАРТОЧКА

ПВ | место | литер

на ~~газету~~ журнал (индекс издания)

Linux Format

(наименование издания)

Подписная цена | руб. | коп. | Количество комплектов:

на 200_ год по месяцам

1	2	3	4	5	6	7	8	9	10	11	12

Куда (почтовый индекс) (адрес)

Кому (фамилия, инициалы)

ПОДПИСКА НА LINUX FORMAT

ПОДПИСКА ПО КАТАЛОГАМ СНГ И БЛИЖНЕГО ЗАРУБЕЖЬЯ

Каталог «Российская Пресса»- совместный проект Государственного предприятия «Казпочта» , Агентства «Книга-Сервис» и АРЗИ.

Блок изданий АРЗИ в национальных Каталогах Украины и Беларуси. В Азербайджане, Армении, Грузии, Киргизии, Узбекистане и Молдове - по изданиям, включенным в Объединенный каталог, распространяемые через АРЗИ.

Азербайджан

- по Объединенному каталогу российских изданий через Предприятие по распространению печати «Гасид»(370102, г. Баку, ул. Джавадхана, 21);

Армения

- по списку номенклатуры «АРЗИ» через ГЗАО «Армпечать» (375005, г.Ереван, пл.Сасунци Давида, д.2) и ЗАО «Контакт-Мамул» (375002, Г.Ереван, ул.Сарьяна, 22);

Белоруссия

- по Каталогу изданий стран СНГ через РГО «Белпочта» (220050, г.Минск, пр-т Ф.Скорины, 10);

Грузия

- по списку номенклатуры «АРЗИ»» через АО «Сакпресса» (380019, г.Тбилиси, ул.Хошараульская, 29) и АО «Мацне» (380060, г.Тбилиси, пр-т Гамсахурдия, 42);

Казахстан

- по Каталогу «Российская Пресса» через ОАО «Казпочта» и ЗАО «Евразия пресс»;

Молдавия

- по каталогу через ГП «Пошта Молдавей» (МД-2012, г.Кишинев, бул. Штефан чел Маре, 134);
- по списку через ГУП «Почта Приднестровья» (MD-3300, г.Тирасполь, ул.Ленина, 17);
- по прайс-листу через ООО Агентство «Editil Periodice» (2012, г.Кишинев, бул. Штефан чел Маре, 134).

Узбекистан

- по Каталогу «Davriy nashrlar» российские издания через Агентство по распространению печати «Davriy nashrlar» (7000029, Ташкент, пл.Мустакиллик, 5/3, офис 33);

Украина

- Киевский главпочтамт.
- Подписное агентство «KSS» Телефон/факс (044)270-62-20, 270-62-22



АЛЬТЕРНАТИВНЫЕ АГЕНСТВА РФ

Агентство «Интер-Почта»
(095) 500-00-60, курьерская доставка по Москве.

Агентство «Вся Пресса»
(095) 787-34-47

Агентство «УралПресс»

- Екатеринбург, Березовский, В. Пышма, Первоуральск
  тел. (343) 375-80-71, 375-84-93, 375-84-39,
  факс 375-62-74,
  info@ural-press.ru
- Нижний Тагил
  тел. (3435) 411448, 417709,
  ntagil@ural-press.ru
- Челябинск
  тел. (351) 262-90-03, 262-90-05,
  pochta@chel.surnet.ru
- Пермь
  тел. (3422) 60-24-40, 60-22-95, 60-35-42,
  parma-press@permonline.ru